

Trust Policy Management for the Financial Industry using Semantic Web Rules

by

Chitravanu Neogy

M.S., Mechanical Engineering, University of Cincinnati, 1991
B.S., Mechanical Engineering, Indian Institute of Technology, Kanpur, 1989

Submitted to the Alfred P. Sloan School of Management in partial fulfillment
of the requirements for the degree of

Master of Science in the Management of Technology

at the

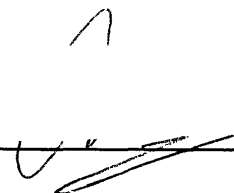
Massachusetts Institute of Technology

June 2004

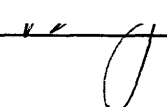
© 2004 Chitravanu Neogy. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute
publicly paper and electronic copies of this thesis document in whole or in part.


Signature of Author: _____

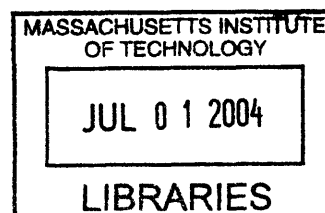

Chitravanu Neogy
Sloan School of Management
May 7, 2004

Certified by: _____


Prof Benjamin Grosz
Sloan School of Management
Thesis Advisor

Accepted by: _____


David Weber
Director, Management of Technology Program



ARCHIVES

Trust Policy Management for the Financial Industry using Semantic Web Rules

By

Chitravanu Neogy

Submitted to the Sloan School of Management on May 7, 2004
in partial fulfillment of the requirements for the degree of
Master of Science in the Management of Technology.

Abstract

Trust Management is a growing problem in large corporations today. In industries like financial services, firms need to comply with constantly changing regulations, security requirements and business policies. Information technology is often the backbone of the processes that are regulated by such policies. Traditionally fine-grained Trust Management has been attempted by embedding policies within business logic of silo software applications. This practice leads to high total costs of ownership, minimal interoperability, potential security vulnerabilities and low management visibility into policy specifications and enforcement, which complicates compliance challenges with regulations like Sarbanes Oxley.

This thesis makes several new contributions. First, it evaluates trust-policy related applications in the overall financial services industry that can benefit from rule technologies. A second contribution is proposing SCLP RuleML, an emerging semantic web rule language, for representing trust policies (SCLP = The Situated Courteous Logic Programs knowledge representation). A third contribution is providing several financial application scenarios in SCLP that demonstrate the effectiveness of RuleML, including credit card authorizations for electronic transactions, Check 21 processing in banks and account access control in brokerage or mutual fund systems. Finally we provide a rationale and a proposal for RuleML to be a reference implementation of eXtensible Access Control Markup Language (XACML), an evolving OASIS standard for digital authorization. Potential benefits of such standardization include lower cost and more effectiveness of policy administration; better governance and coordination through centralized ownership or interoperability; and reduced system development costs over the full life cycle.

Thesis Supervisor: Prof. Benjamin Grosz
MIT Sloan School of Management

Dedication

This thesis is dedicated to my mother, Mrs. Krishna Neogy, who left us forever in 1993. From my high-school age, she had always wanted me to go to MIT for higher education but somehow I had never pursued that wish during her lifetime. It is my greatest sense of achievement that I diligently remembered her persistent encouragement for so many years was finally able to see the wisdom of her thoughts when I joined MIT as a student. The thesis brings closure to a story of both hope and challenge between a mother and a son.

It is my sincere hope that readers of this research document will spare an idle moment to reflect upon the intense love a mother possesses for her child, her passion to see him succeed in life and the powerful, lasting influence that she has on the destiny of her children.

Table of Contents

1. INTRODUCTION	6
1.1 MANAGING TRUST IN DIGITAL APPLICATIONS	6
1.2 DEFINING AND BUILDING THE SEMANTIC WEB.....	8
1.3 OUTLINE OF THIS THESIS	13
2. CHAPTER 2.....	16
2.1 CHALLENGES FOR TRUST POLICY MANAGEMENT IN THE FINANCIAL INDUSTRY	16
2.2 BUSINESS DRIVERS FOR MAJOR TECHNOLOGY CHANGES	20
2.2.1 Consumer Householding.....	21
2.2.2 Paperless Checks	22
2.2.3 Straight-through processing (STP)	22
2.2.4 T+1 processing.....	23
2.2.5 Basel II.....	23
2.2.6 Sarbanes Oxley Act of 2002.....	23
2.2.7 Gramm-Leach-Bliley Act of 1999 (GLB)	24
2.2.8 Health Insurance Portability and Accountability Act of 1996 (HIPAA).....	25
2.2.9 US Patriot Act of 2001.....	26
2.3 BUSINESS AUTHORIZATION – DEFINITIONS AND PERCEPTIONS	29
2.4 BENEFITS OF STANDARDS TO IT PLAYERS	33
3. CHAPTER 3.....	38
3.1 REVIEW OF SEMANTIC WEB RULES TECHNOLOGY	38
3.2 ADVANTAGES OF SEMANTIC RULE-BASED TRUST POLICY MANAGEMENT	41
3.3 TRENDS IN THE BUSINESS RULE ENGINE MARKET	42
4. CHAPTER 4.....	47
4.1 ADVANTAGES AND CAPABILITIES OF SEMANTIC WEB RULES TECHNOLOGY FOR TRUST IN FINANCIAL SERVICES INDUSTRY	47
4.2 CONSUMER HOUSEHOLDING	48
4.3 COMPLIANCE REGULATIONS	52
4.4 SECURITY AUTHORIZATION AND XACML.....	55
4.5 FAST MOVING MARKETS.....	59
5. CHAPTER 5.....	63
5.1 EXAMPLES OF BUSINESS RULE BASED AUTHORIZATION IN THE FINANCIAL INDUSTRY	63
5.2 CASE IMPLEMENTATIONS USING SEMANTIC WEB RULES	63

5.2.1 Case I: Credit Card Verification System for Electronic Transactions	64
5.2.2 Case II: Check Clearing for the 21 st Century Act.....	69
5.2.3 Case III: Brokerage System Account Access	72
5.2.4 Conclusion from SCLP examples.....	78
6. CONCLUSION	80
6.1 RESEARCH FINDINGS	80
6.2 SUGGESTED FUTURE WORK	84
7. ACKNOWLEDGEMENTS	87
8. REFERENCES AND BIBLIOGRAPHY	89

1. Introduction

“No unit of information is too basic to prevent disagreement about its meaning”

-Thomas H. Davenport [7]

1.1 Managing Trust in Digital Applications

Trust encompasses a broader scope and deeper meaning in business environments than is apparent from its literal meaning. Mayer et al [33] defines trust as the “willingness to be vulnerable to the actions of another party”. Another study characterizes trust as “the probability one attaches to the cooperative behavior by other parties” [23]. Luhnman [32] sees trust as the belief by one party about another party that the latter will behave in a predictable manner. Many people believe that trust entails a perception of risk [5] and almost everyone agrees that trust is important in business relationships, especially in e-business [14] because it has been shown to affect the adoption of new technologies, including the World Wide Web.

In electronic services, Trust is often synonymous with security. If two parties can rely on each other, they can conduct business together. In a conventional world, people build trust through relationships. However, in cyberspace, many trading parties or systems may not have a *prior* relationship with each other and yet may need to negotiate on specific deals. Third-party references are often essential as an avenue to establish trust in such cases. This raises the question of whether

the independent third party is trusted equally by participating business entities. Then we have to deal with “shades of Trust”, in lieu of the question – “How much do we trust a particular entity”, and the answer could cover a wide range of possibilities. To manage risks under such circumstances, firms in practice perform only certain business activities for each range of trust realized.

Trust Management is even harder in a distributed environment [2]. Here a set of credentials attached to an accessor has to be evaluated by an authorizer to ensure that the request complies with local policies [30]. As mentioned earlier, the authorizer may have had no prior knowledge of the user. Credentials may be facts, or more generally, non-local policy statements. Generally since credentials are not always under the control of the authorizer, they need to be tamper-proofed, often using digital signatures. TM systems also need to delegate permissions from the credential issuer to its subject.

Online Trust has several possible antecedents and consequences [51]. In a typical trading scenario, such antecedents may include factors like the seller’s reputation, relationship-specific investments by the seller, size of his firm, economic outcome of the trade, prior buyer experience with the seller and the incidence of opportunistic behavior demonstrated by the seller. Satisfaction and long term orientation are the primary consequences of trust.

In order to define authorization policies and credentials in distributed web applications, we need a Trust Management language [21, 9]. Such a language must meet certain explicit requirements in terms of expressive power, declarative semantics and tractability [30]. Several TM systems have been proposed in recent times including SPKI, PolicyMaker, KeyNote and Referee – however, none of these languages possess all the core requirements needed from a TM. Grosz et al. posits that the problem is largely one of Knowledge Representation (KR) and is in part well solved by a logic programs approach. They propose a new construct called D1LP (Delegation Logic) as a practical Trust Management language [21, 30].

In this thesis, we will focus on Trust Management Rules, using an emerging standard called RuleML [19, 43], which is based on the Situated Courteous Logic Programs (SCLP) [21, 17]. RuleML allows exchange of rules through XML. A significant contribution of this work will be to explore how SCLP can be used to develop scalable trust management applications for solving a wide variety of complex problems in the financial industry [17].

1.2 Defining and building the Semantic Web

Semantic web allows search agents, information brokers and information filters to offer much richer functionality than stand alone systems [34, 28]. Perhaps the

best description of the semantic web comes from Tim Berners-Lee, the founder of the World Wide Web:

The Semantic Web is not a separate web but an extension of the current one...Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully...The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.

Tim Berners-Lee et al, Scientific American, 2001 [28]

In his book *Understanding Web Services*, [38] Eric NewComer states that web services allow applications at different network locations to communicate “as if they were part of a single, large software system”. Agents at different locations can communicate automatically and seamlessly with their cohorts. As a result of the new agent-driven framework, we can expect to see a shift from hands-on browsing to hands-off delegation to “black box” services. Security of the infrastructure and the agents [10, 11] play a very important role in ensuring the safety and reliability of web services and much work [39, 26] has already been contributed to defining such requirements.

The global vision of the Semantic (or meaningful) interoperability is supported by several emerging IT standards including Extensible Markup Language (XML), Resource Definition Framework (RDF), Ontologies, RuleML and Web Services.

XML addresses “only document structure” like web page annotation, whereas RDF [8] represents metadata about web resources through XML. Ontology defines the terms that are used to describe an area of knowledge [6]. A significant standard language in ontology is Ontology Web language (OWL), which had DAML+OIL [6] as a close predecessor. This in turn originated from DARPA Agent Markup Language and Ontology Inferencing Language. RuleML is an emerging XML-based standard for representing rules using a logic language called Situated Courteous Logic Program (SCLP) [21,17]. Figure 1 shows the emerging standards in different components of the Semantic Web development wave [27].

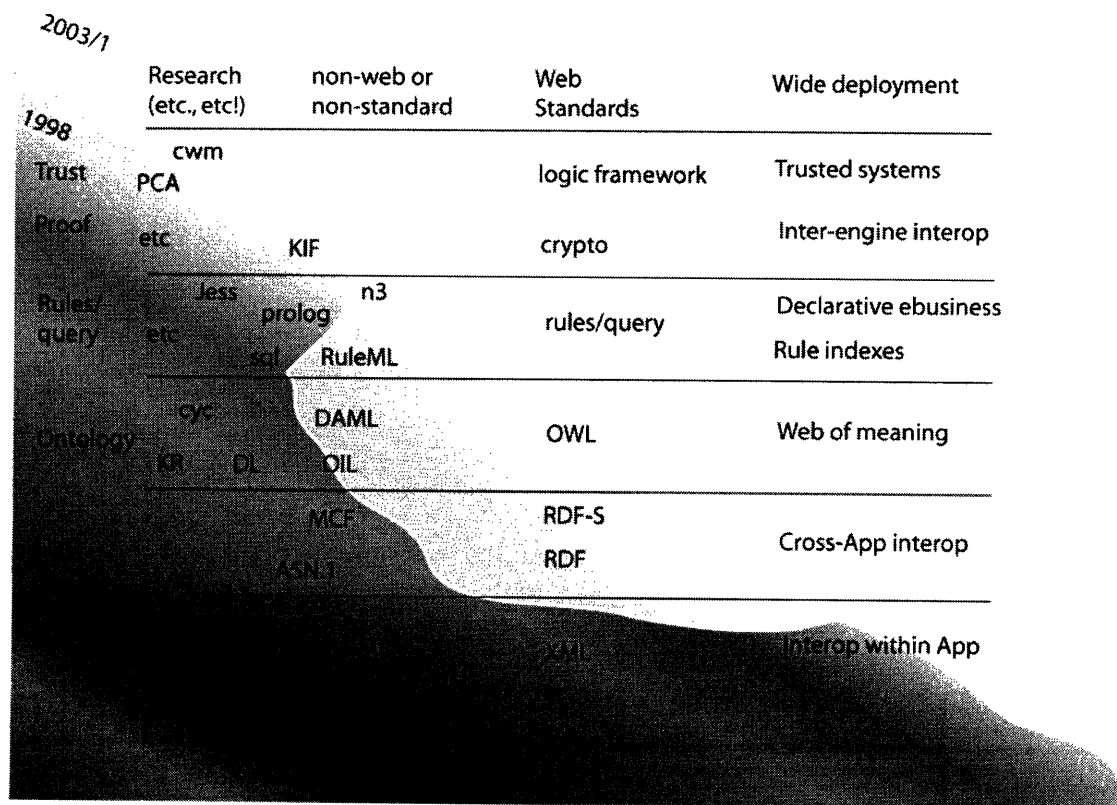
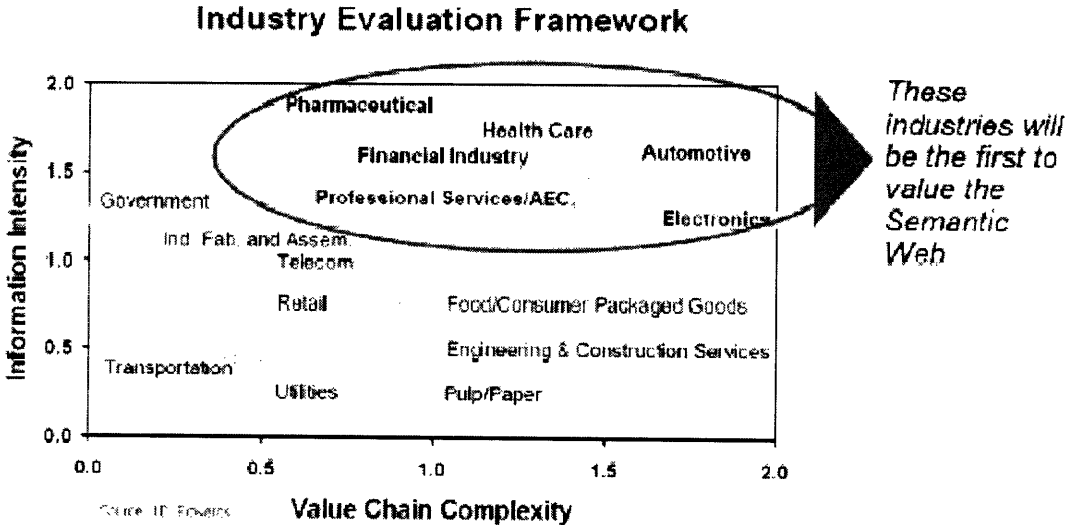


Figure 1. Technology Components of the Semantic Wave Value Chain [28]

In the near future, we can expect to see intelligent agents in all major business industries including transportation, credit cards, equity trading and energy distribution. Grosf et al. talks of personalization capabilities [30] in bookstore discounts, supply chains or retail refunds. Another example is in stock trading, where the best an investor can do today in terms of automation is to place limit orders for trade executions. In the future, automated software agents acting on behalf of human investors can time the market much better by following news releases and devising an optimal investment strategy on the fly. Gazis demonstrates this concept through a Personal Application-Specific Hyper-intelligent Agent (PASHA) is being designed to perform personalized information search but also can perform tasks according to a user wish list.

Based on industry adoption trends, it appears that Web Services will be the next generation in the web's evolution. Web Services work through standards like Simple Object Access Protocol (SOAP), WSDL and directory services like Universal Description, Discovery and Integration (UDDI) [29, 38]. The goal of web services is similar to that of the semantic web in making the internet more "machine processible". Preece and Decker [42] accentuate this intersection, saying that as the descriptions of agent-based web services are updated in the directory, a higher value will be created through client services extending themselves to match the new capabilities of the service.

Industries that depend heavily on rule-based processing and manage large amounts of information as part of the core business will likely be early adopters of the Semantic Web [18,25]. Kabbaj, a graduate student of Grosf suggests that News Syndicates, Portals, Technology consultants etc. are likely to be the lead users. The financial services, electronics industry and pharmaceuticals with high complexity across the value chain will likely drive the next wave of adoption, followed by government, telecomm and retail sectors.



... the semantic web allows computers to understand complex products, to operate in complex value chains and to lower transaction costs.

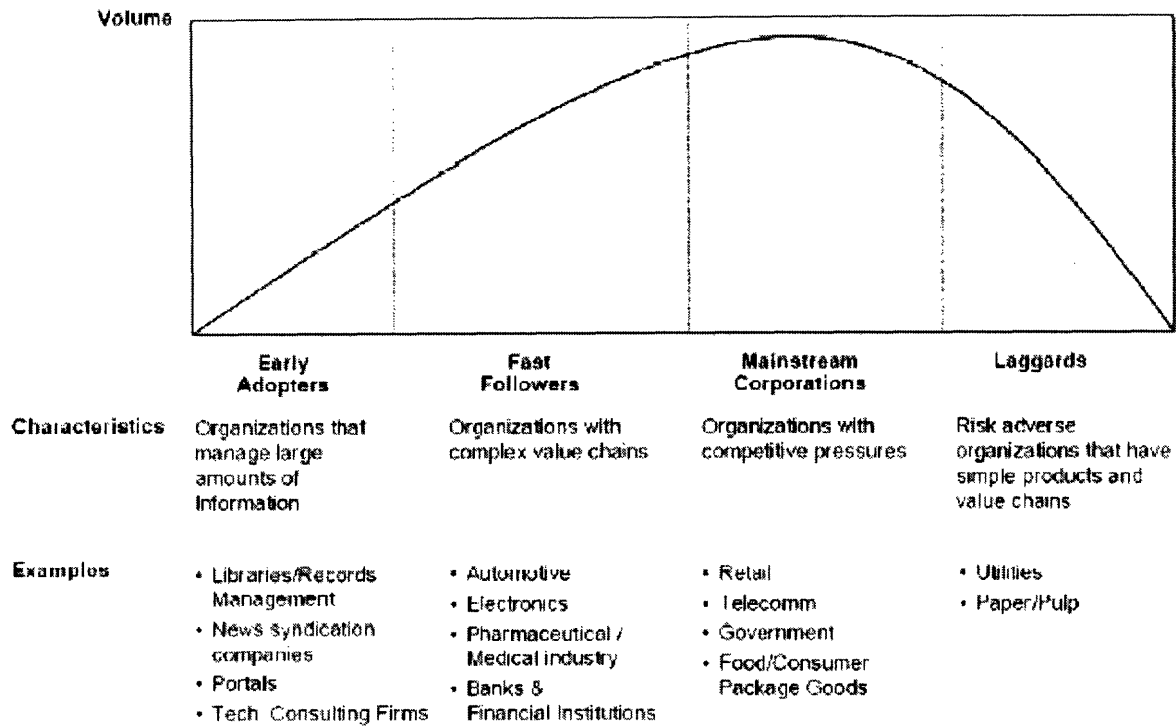


Figure 2: Semantic Web adoption rate predictions

1.3 Outline of this Thesis

This thesis is organized in the following manner. The introduction provides an overview of trust concepts in the electronic world, especially in distributed web applications. We also discuss the various components of the Semantic Web, evolving standards surrounding its implementation efforts and the markets where this technology may generate significant benefits.

In Chapter 2, we talk about financial services industry and their IT spending habits, especially for retail and investment banking, retirement accounts, asset management, brokerages and mutual funds, etc. Next we discuss major trends in

the financial services business and describe IT initiatives that correspond to these changes, many of which are in response to regulatory acts like Patriot Act or Sarbanes Oxley. We highlight the challenges and risks of implementing each such system, including limited reuse and inflexibility of maintaining policy applications or Role Based Access Control (RBAC) user profiles. This led to a definition of Business Authorization, where access policies are based on user identity and application data or rules. In Chapter 2, we also develop an argument in favor of employing standardization to benefit the vendors, programmers and users of software products.

In Chapter 3, we study Semantic Web Rules Technology and compare leading CCI rule systems with RuleML [17, 19], an emerging standard in rules knowledge representation. We discuss the advantages of representing Trust Policies in a standardized rule language. An overview of leading Business Rule Engine (BRE) vendors and BRE technical requirements is provided as well.

Chapter 4 integrates the discussion in earlier chapters to expound upon the advantages and capabilities of using semantic web rules for Trust Management in the financial services industry. We discuss the applicability of rule technologies in several business challenges like Consumer Householding, Check 21, Regulatory Issues, Security Authorization (XACML), Program Trading and Email Filtering.

In Chapter 5, we provide some application scenarios of RuleML for knowledge representation of rules in business problems like credit authorization, “Check 21” processing and brokerage account access control. Since RuleML tools are still immature, we have used IBM Commonrules, an SCLP toolkit to represent the business policies for these situations.

Chapter 6 is the Conclusion, where we have summarized the research findings. We posit that standards based rule technologies like RuleML can break significant market barriers and accelerate adoption of this technology, thereby benefiting BRE vendors. We suggest a number of future research activities including development of RuleML compliant software and applying similar analysis as in this thesis to other areas within financial services like the insurance or mortgage industry.

2. Chapter 2

2.1 Challenges for Trust Policy Management in the Financial Industry

The Financial Industry in the United States is comprised of many diverse markets. Commonly known examples include Consumer Banking, Investment Banking, Credit Card, Mortgage, Brokerage/Mutual Funds, Insurance, Retirement, Venture backed financing and Private Asset Management. To underscore the impact of this industry on the economy, just the mutual fund industry in US is a 7 trillion dollar market. The commercial insurance business size in USA is about \$120B.

Consumer or retail banking involves normal customer account services including checking, savings, money market accounts, personal loans, wire transfers, bill payments etc. Most banks and other financial institutions also offer branded credit cards today for both individual and business purposes. Each such company is part of the member network of a payment services organization like Visa or MasterCard. Mortgage companies offer home or business loans for the housing market, equity line of credit and debt consolidation services. Insurance is a large risk-management industry, covering almost anything valuable from personal homes, vehicles, jewelry, business assets etc. Brokerage houses perform sale of stocks of publicly traded firms in the secondary auction markets (NYSE) or over the counter exchanges (NASDAQ). Mutual fund companies operate funds in

which they buy equities of firms using capital provided by individual or institutional investors, with objectives of higher than market returns.

Investment banking involves businesses specializing in the formation of capital. Such banking firms act as underwriters or agents for companies issuing securities, and advise the company on the issuance and placement of its stock. Investment bankers assume the risk of selling the securities of companies to its customers. They also participate in takeover financing, including Mergers and Acquisitions.

Retirement accounts can be of multiple types and are subject to stringent government regulations, especially regarding benefit eligibility and tax provisions. In this business, professional asset management companies manage the entire employee retirement savings of client companies. Since larger companies have many employees (e.g. GM has more than 100,000 associates worldwide), winning a blue chip firm's retirement business can be very profitable. There are two common types of retirement accounts:

- (a) Defined Benefits – a.k.a. pension funds, where the customer receives a fixed benefit amount after retirement eligibility date, for the duration of lifetime. These are typically fully managed services and all contributions for the fund come from the employer. The company managing the funds on behalf of the customer collects a management fee on a per seat basis.

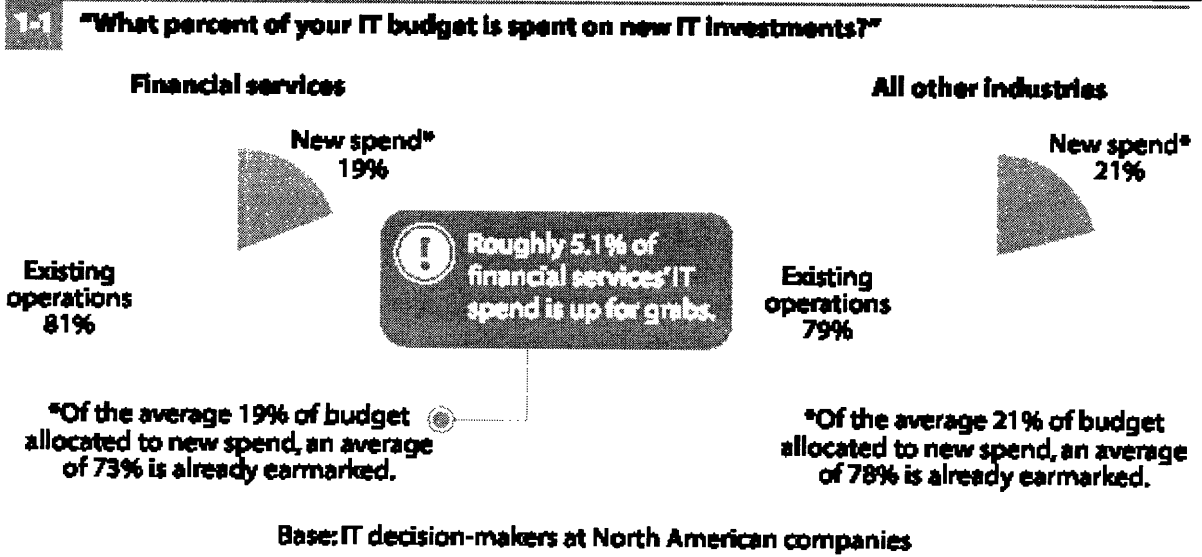
(b) Defined Contributions – In this more popular practice, consumers are given much greater control of their retirement accounts. 401K is a type of Defined Contribution. Customers determine themselves how much of their monthly before-tax pay they would like to withhold in the 401K fund and companies may or may not match a part of the contribution. Typically, clients can also direct their periodic payments into different type of mutual funds as governed by management company's policies.

Venture Capital firms participate in deal flows for infusion of money into new companies or projects. They have very high annualized return requirements within a short time frame, typically within 5 to 7 years. On other hand, Private Asset Management firms seek long-term capital appreciation of wealthy individuals. They typically apply analytical processes to management of client portfolios and need to generate significant above market returns for investors.

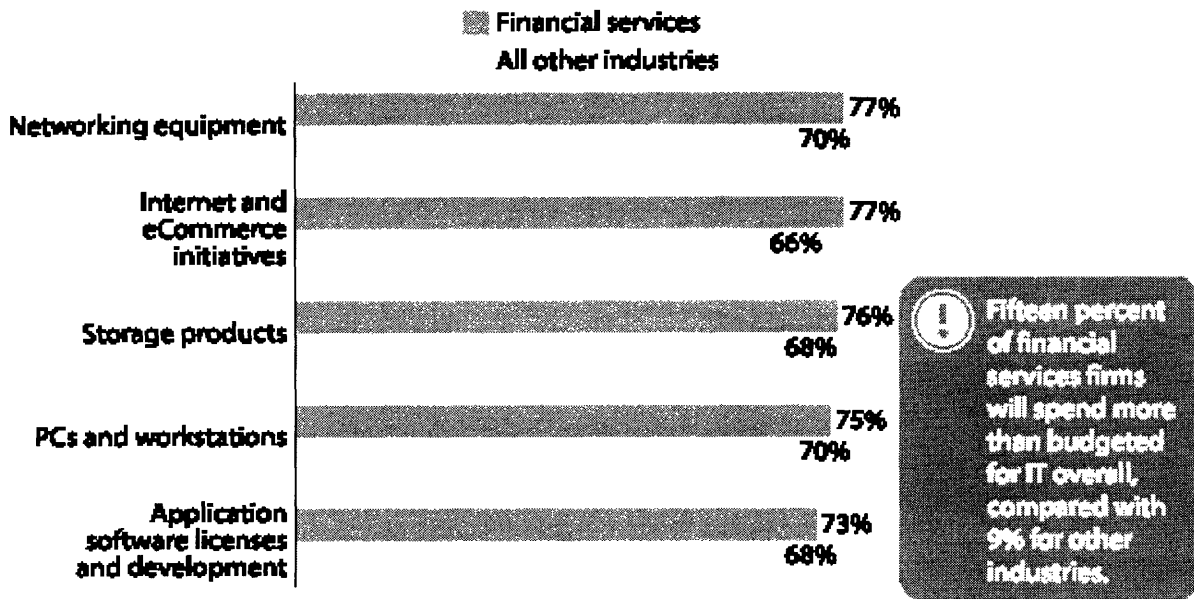
There are also companies that provide aggregate services, combining any two or more of the above broad sectors. For example, Fleet offers retail and business banking but also partners with Quick and Reilly to offer brokerage services to its clients through the same channel. Similarly Fidelity Investments plays across the financial services value chain, including trading, mutual funds, consumer banking, institutional investors, retirement benefits,

Partly because the stakes are so high and the industry deals directly with money, federal and government regulations remain an integral part of the financial business to ensure fairness, ethical business practices, consumer protection and privacy. The traditional financial services industry has been totally transformed by technology advancements over the past decade. A large part of this change is from new business models and unparalleled efficiencies achieved through IT developments. Some of the larger Financial Service Providers (FSP) spends almost 20% of their annual budget on IT investments. The 2003 Forrester report [13] on financial industry technology spending predicted that 27% of *new dollars* within financial firms is up for grabs, not yet earmarked for specific projects. The following chart shows a comparison of IT spending in financial services versus other industries. (the figure number refers to that in the original report)

Figure 1 The State Of 2003 IT Spend And Staffing For Financial Services Firms



1-3 Percent of firms expecting actual IT spend during 2003 to be on plan



Base: IT decision-makers at North American companies

Figure 3: IT spending predictions and patterns in financial services

2.2 Business Drivers for Major Technology Changes

Technology spending in financial services is expected to remain high over the next few years because of several industry drivers, as summarized below [36]. In this section, we will describe some of these key stimulants and discuss their impact on financial services firms.

The following table lists some of the expected business needs in the financial industry over the next two years. Each of these requirements will drive management to employ one or more software-based solutions. Overall, the major

spending will occur in projects that bring about consolidation of IT resources, meet regulatory requirements or standards and build competitive advantage.

Business Drivers	Technology Impact
M&A, Householding, Consolidation and Simplification of computer systems to reduce costs.	Need for Enterprise Integration, CRM, BPM, ERP (like SAP), Web Services
EDI, Paperless checks (Check 21), Straight Through Processing (STP), T+1 processing	More intra-firm and inter-company technology standardization, greater automation, higher throughput
Regulations (Sarbanes Oxley, GLB, HIPAA, Patriot Act, Basel II initiative), Identity Management [37]	Require Pervasive security and audit controls, Employee Education and compliance software, digital signatures, IPV6, Rule Engines
Trading Systems, Wireless Services, P2P applications, real-time analytics	Performance, HA, Security, Data Mining and Modeling Techniques
Flexible systems in fast moving markets, e.g. program trading, margin rules, antivirus protection policies	Rule-based Architecture, Business Activity Monitoring (BAM), Web Services

Figure 4: IT challenges in the Financial Services Industry

2.2.1 Consumer Householding

Better Customer Relationship Management (CRM) within financial institutions requires aggregation of client asset information. Many companies want to see a consolidated view of not just a single customer's accounts, but also that of his entire household, e.g. spouse, children, parents etc. This will establish the real net worth of that individual to the firm and allow for delivery of additional services important to the complete customer family. For example, a person may currently own marginal holdings in a bank but may be the son of a very high net worth

couple. With Householding capabilities, firms can recognize greater ROI by effectively segmenting client profiles, delivering targeted, more effective ads and reducing customer contact fatigue through customized statements and newsletters.

2.2.2 Paperless Checks

The Check Clearing for the 21st century act (Check 21) facilitates significant changes in the ways checks are processed in the United States [3]. The law revolves around two concepts (a) check truncation and (b) electronic check presentment. The act creates a new negotiable instrument called the substitute check that is legally equivalent to a paper check if it meets a set of requirements laid down by the Federal Reserve Board. Today, checks are trucked from bank to bank after they are deposited, till it reaches the payer's bank. Check 21 legislation is expected to save the banking industry billions in transportation and float related costs. The law goes into effect on Oct 28, 2004.

2.2.3 Straight-through processing (STP)

STP refers to the goal of the securities industry to create a trade-processing environment in which trades are conducted seamlessly among all involved in the trading process, without any manual intervention or redundant processing. This primarily involves streamlining computer operations and replacing manual processes with digital applications.

2.2.4 T+1 processing

This represents the reduction of the standard period between trade date and settlement date from three days to one day. The scope of the T+1 conversion will impact all corporate equity and debt securities, some municipal bonds, and US Government Agency securities (non-mortgage backed). Many of the issues surrounding implementation of an STP system coincide with the move to T+1. Because of significant challenges in implementing T+1 by mid 2005, the original deadline, the Securities Industry Association (SIA) has postponed this requirement till STP is achieved across most firms.

2.2.5 Basel II

The Bank for International Settlements (BIS) initiative affects all European banks, as well as some of the larger US banks [54]. The European Basel accord consists of three pillars (a) minimum capital requirements (b) supervisory review of capital adequacy (c) public disclosure. The overall idea is to prevent some banks from going suddenly bankrupt, causing financial losses to thousands of depositors.

2.2.6 Sarbanes Oxley Act of 2002

It is the hottest regulatory topic in most US corporations today [45]. This sweeping regulation came at the wake of major accounting frauds in publicly-trusted large

companies and is aimed to protect shareholders and employees from losses and corporate scandals. The act mandates stringent policies revolving around Board Membership and regulations, registrations, inspection of CPA's, security of documents for later investigations, accounting standards, auditing practices, corporation taxes etc.

While there is a need for greater corporate governance in the aftermath of Enron Corporation, Tyco International and MCI WorldCom, minding the shop is becoming excessively costly, especially for middle market companies. Surveys have reported an increase of 90.4% in the cost of being a public company, a daunting number that is causing some companies to de-list from the exchanges. A ComputerWorld article [44] indicated that the IT costs for Sarbanes Oxley will approximate \$500,000 per firm. AMR Research estimates that Fortune 1000 companies will spend almost \$2.5B in 2003 on S.O. compliance. Exorbitant hidden costs to the tune of millions have also been predicted for leading firms. However while the compliance requirements are steep, most companies feel there are no other ways around it if public faith is to be restored.

2.2.7 Gramm-Leach-Bliley Act of 1999 (GLB)

The Financial Modernization Act of 1999, also known as the "Gramm-Leach-Bliley Act" or GLB Act [53], includes provisions to protect consumers' personal financial information held by financial institutions. There are three principal parts to the

privacy requirements: the Financial Privacy Rule, Safeguards Rule and pretexting provisions.

The Financial privacy rule governs the collection and disclosure of customer's information by the FSPs. The Safeguards rule requires all financial institutions to implement policies, systems etc to protect the privacy and confidentiality of customer information. Pretexting provisions of the GLB Act protect consumers from individuals and companies that obtain their personal financial information under false pretenses, a practice known as "pretexting."

2.2.8 Health Insurance Portability and Accountability Act of 1996 (HIPAA)

Also known as the Kassebaum-Kennedy Act [22], this law aims to help people keep their health insurance, even if they have serious health problems. It also mandates administrative, privacy and security requirements of health plans and seeks to reduce the overall cost of healthcare. While this bill primarily affects medical service companies, financial institutions (and others) with employees registered in company sponsored health plans must meet certain compliance guidelines.

2.2.9 US Patriot Act of 2001

This bill was spawned after 911 to deter and punish terrorist acts in the United States and around the world [55]. The act gives federal officials greater authority to track and intercept communications, both for law enforcement and foreign intelligence gathering purposes. It provides special powers to combat money laundering in financial institutions by terrorists. It creates new crimes, new penalties and new procedural efficiencies for use against domestic and international terrorists. From a financial organization's viewpoint, the law requires filing of Suspicious Activity Report (SARs), due diligence measures for anti-money-laundering, prohibiting US institutions from maintaining correspondent accounts for foreign shell banks, enforcing visibility into customer's financial activities, establishing new customer identification standards, encouraging greater cooperation between financial companies and law-enforcement agencies, etc.

The rapid pace of technology adoption, increased customer sophistication, higher incidence of frauds, M&A activities and a greater number of regulatory issues have led to significant difficulties in maintaining electronic security and enforcing business policies. Integration of systems across the enterprise is an ongoing challenge. Part of the problem in financial services is because of the many legacy mainframe applications that comprise the majority of back-end transactions. New technologies like Web Services can ease the pain of interfacing with such proprietary systems. Vendor battles have somewhat retarded the immediate

adoption of Web Services but in the longer term, this technology is expected to deliver the highest improvement in interoperability problems.

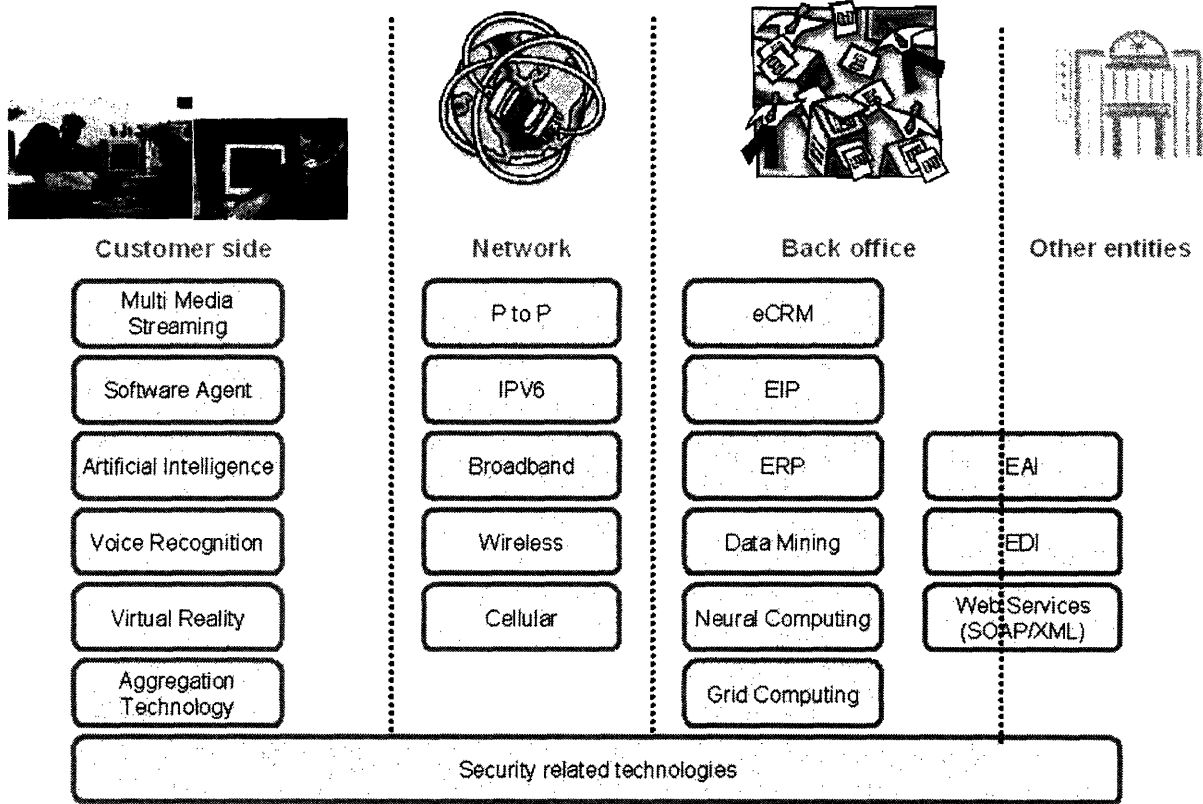


Figure 5: Technology infrastructure across enterprise applications [36]

Trust Policy Management is inherent in all these business efforts and complexities of security integration continue to expand as the number of enterprise applications increase. Policies in financial services typically tend to be one of the following:

1. Federal/State Mandates (SEC, GAAP, Central Bank requirements)
2. Legislative regulations (Gramm-Leach-Bliley Act, Patriot Act, Anti money-laundering, HIPAA, Sarbanes-Oxley etc)

3. Financial Industry Specific Policies (Blue Sky rules, Series 7 registration, Liberty Alliance)
4. Firm Specific Procedures (Margin account requirements, access policies, retirement benefit stipulations, fiduciary responsibilities etc)

Interviews during the research indicates that a firm with \$75B in assets in under management may have to manage and comply with 10,000 to 20,000 rules, many of which will be implemented at a system level. This is indeed a complex task, especially since most rules are interpreted by a few legal experts or business analysts, but need to be developed and maintained by IT personnel. Without flexibility and a standardized rulebase, most policies end up being coded directly into silo applications.

Such a system has limited reuse or flexibility, thereby leading to redundant functionality between applications and relative obscurity with relation to the meaning or enforcement of embedded policies. Developing software is an expensive proposition. To make the ROI attractive, companies should consider using standardized components as development tools so that (a) less specialized skills are required to maintain the product and (b) other units developing a different application with similar business rules and common modules can reuse already validated code to save precious time and money for the organization. For example, within the retirement business, benefits eligibility rules are often

federally mandated and quite similar across services, whether its 401K, 403B, IRA, pensions or retirement savings accounts. All of them would compute the allowed retirement age for benefits collection based on Social Security age. If these are all offered services within a firm, it is likely that benefit calculations would be based on similar elements (e.g. vesting requirements, last compensation, interest rate etc.). Accordingly, sharing these data and their access methods between all applications make sense. A similar illustration involves self-account lookups by a customer. Most applications allow clients to look at their own assets online, yet this capability either has to be hard-coded in a standalone piece of software or represented in unwieldy profiles per user, using Role Based Access Control (RBAC) guidelines. Yet, a standardized logic program can handle the same requirement easily through a simple rule.

This premise leads to the need for establishing a standardized Trust Management paradigm for encapsulating the “Business Authorization” mechanism.

2.3 Business Authorization – Definitions and Perceptions

The term authorization in information systems context is often related to network security. In most enterprises, cyber-security has 4 components [24]:

- Authentication – verifies principal is as claimed
- Authorization – determines access rights for an authenticated principal

- Administration – assigning access capabilities to different users, groups or resources
- Audit – ensuring that system access incidents, failures and violations are logged and available for playback and audit later

Authorization itself can occur in at least two levels:

- (1) Transactional authorization: Certain transactions can only be run by privileged individuals. For example, a service representative in a mutual fund company can perform account views or margin allocations, but cannot run a trading transaction on a customer account. Generally, RBAC model is quite useful in mapping access policies for these transactions.
- (2) Data Authorization: This is a finer level of access control where a user may be empowered to run transactions for a limited set of data. For example, if I work in Federal Street branch of Fleet, I may be allowed to look at any customer accounts that were initiated in my bank, but cannot view accounts from other Fleet branches. This is because there is a bank policy that prevents cross-branch account access even within the same lookup transaction. In real life, there may actually be more rules that determine the account access capabilities of a user; such as if I am a Regional Manager, I may enjoy access to accounts in my home branch plus a list of other branches under my control. These are all examples of Data Authorization.

Data Authorization often falls in an overlapping area between infrastructure security and business applications. For example, in the previous account lookup case, we needed user, role and location information that could have come from a centralized security database. On other hand, information about branch hierarchy and account-branch relationship is from business repositories. We need both pieces of information to make an access control decision, viz. should a person be able to look at a certain customer account. We define this type of scenario as a case of *Business Authorization*, a key area of focus in this thesis. The following diagram aims to represent this concept.

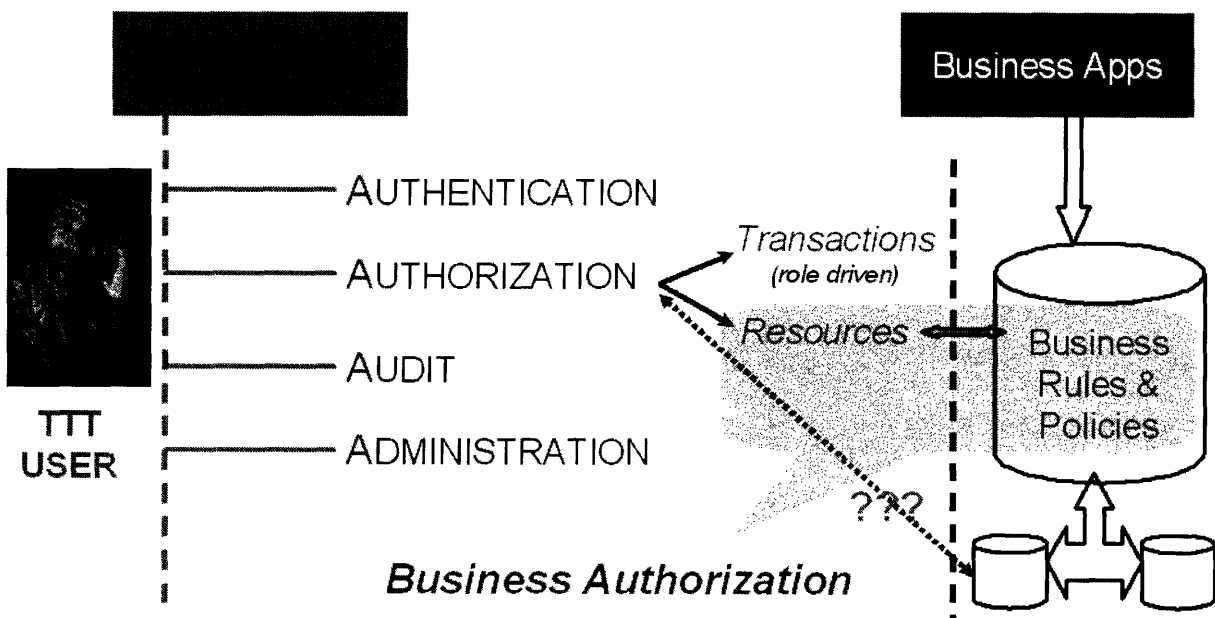


Figure 6: How we define Business Authorization

Today, in the financial services world, business authorization really takes place within the individual applications, as opposed to being an integral part of the

security architecture. In general, implementation of business authorization is a sticky point in many architecture discussions for many reasons:

- a) Complexities of ownership – Questions arise whether this is part of security or a business application. There is a technical challenge as well as a political consideration in placing Business Authorization in either bucket. Since information from both areas must be integrated together to make a certain authorization decision (e.g. can I look at this account), some data exchange must take place within an application context, and there can be technical concerns about how that is achieved in practice. We often see negotiation regarding which party provides data (user ID, account number, branch code etc.) versus who executes the rule to make a decision.
- b) Skepticism whether complex business rules can be expressed uniformly in a security paradigm
- c) Will there be performance issues unless authorization decisions are taken within business applications? Can there be confidence in rules enforcement if the business has lesser direct control in implementation?
- d) Power Politics: The value of a businessperson or analyst is often based on the knowledge of embedded policies in legacy applications. Security through obscurity is actually job security!
- e) More “Finger Pointing” in Product support: When customers call for support on a problem, there is more possibility that application support will diagnose the problem as a security issue and vice versa.

Notwithstanding these challenges, there are some advantages of centralized Business Authorization, especially if done on a standardized rule-based platform. As we will see later, key benefits include interoperability, flexibility and system reuse. Development and maintenance costs are lower if a single Trust Management language can be used across the enterprise. It reduces the risks of regulatory burden on firms, since senior management can have much greater confidence that policies are uniformly applied across all IT systems. Many of these benefits accrue as a consequence of emerging dominant designs [56] in the industry which eventually leads to standardization. In the next section, I will discuss the benefits and perceived obstacles to standards technology industries.

2.4 Benefits of Standards to IT Players

According to tradition, the first modern-day standards were developed for railroad gauges and fire hose threads [35]. People got tired for transferring loads from one train to another when track widths changed and they got tired of seeing buildings burn down because the fire hose did not fit the hydrant. Standardization relates to unification, simplification and consistency of technique [12], which often accompanies the installation of a new technical system. In fact, standardization and innovation are often complementary in knowledge-based economies like the US. This is an overall benefit of standardization and occurs because of the following reasons:

1. By solidifying evolution, standards reduce the uncertainty of stakeholders and encourage them to allocate resources and develop technologies that are part of the standard.
2. Standardization improves diffusion of innovation and effectiveness of R&D
3. In some cases, standardization can itself be the innovation, as was when the American Manufacturing System was installed.

A fair amount of research work has gone into exploring how standards in information technology benefit all the players. Bird [1] argues that there is potential benefit for buyers, software authors and software vendors. First, the buyers and users of open standards IT systems are better off because of:

(a) Increased flexibility – the ability to move applications and data from one system to another is vital. Information must not only be transferable between systems, but it must be retrievable in real time to generate business value. Heterogeneity in business computing is the normal case and open systems are critical to bridge differences between the 12+ operating systems that US businesses typically have within their organization. As an example, the Web itself exists because of open standards.

(b) Freedom of Choice – Buyers will have greater power in vendor selection. They can now choose to have best of breed solutions from different vendors and still expect such products to work together in a *heterogeneous* environment.

(c) Lower Integration Costs – Allows purchase of off-the-shelf software components and have them *interoperate* together.

(d) Simpler Purchase Process – If products are standards based, its pretty easy to ensure that requirements are met while buying, rather than if proprietary standards were involved, requiring greater due diligence and feasibility analysis. Evaluation of subsequent bids is also easier.

(e) Greater Competition – By promoting competition, standards can lead the way for greater innovation, as noted earlier.

Standards create value for software authors and integrators as well. First, developers can focus on competitive advantages and uniqueness of their product or service, rather than worry about the language or style of the code. System consultants will not need to spend redundant time trying to figure out interoperability between two software components.

Bird also suggests that system vendors often overlook a key benefit of standardization, i.e. huge market expansion. As more users are lured into using open systems products, new uses of the technology are discovered creating a scale of opportunities, which proprietary solutions can never achieve in practice. The common fear of commoditization is often without foundation, as is evident from the example of the auto industry. Despite external similarity between most

cars, the brand, reliability and performance are really the qualities that users value, not minute differences in technical specifications.

Payback from standards adoption can be swift and significant. Organizations like NASA, having adopted standard IT policies, have dramatically reduced the procurement cycle, increased business flexibility, enhanced agility and brought the latest technology to its people.

Standardization is sometimes seen as a threat to Intellectual Property Rights (IPR) protection [47]. In the very broadest sense, standardization and IPR have the same economic objective – to ensure that society benefits from innovation. However, while standardization encourages a common platform approach to diffusion of innovation, IPR seeks to prolong the life of an innovation once it has been discovered, thereby providing incentives to the innovator to invent new technologies. IPR licensing provisions do make the problem easier, in stimulating creation of more uses for a technology, while allowing the original innovator to garner financial benefits of the product. When standards have substantive overlap with inventions protected by IPR, participating organizations must be willing to give up some of the legal protection on the technology to achieve the greater reward of an expanding market. There is always the danger that standardization may lead to lesser vigilance for the classical “attack from below” [4] from new innovators leading to the death of a mature technology and ultimately the

emergence of a new standard. However, such transitions are generally impossible to prevent through IPRs and occur because of some inherent benefits delivered by the attacker; hence that consideration should not inhibit the decision to license an attractive technology. While there are legal methods for enforcing licensing agreements, in practice standard setting is viewed as a collaborative negotiation process between producers and regulators. Members of a standards committee overcome conflict by agreeing to license technology on fair, non-discriminatory terms. For example, standardization has become critical in the telecommunications industry following market liberalization, leading to an outbreak in the number of firms competing in a level field.

All these insights on standardization are fully applicable to the business rules industry, which is the area of our interest. As we will see later in Chapter 3, there are at least 25 rule vendors competing with proprietary technologies in a narrow client domain and even more producers in the Business Process Management (BPM) space who integrate with rule engines. The user community sees limited value in products of these firms, vis-à-vis the cost of installation, maintenance and integration. Standardization would immediately lower the barriers of rule-engine adoption by reducing total cost of ownership through greater interoperability, lower training and product support expenses. This is the objective of my thesis, through the proposal of RuleML as the business standard for developing rule applications.

3. Chapter 3

3.1 Review of Semantic Web Rules Technology

The standardization agenda of the Semantic Web was represented below by Tim Berners-Lee of W3C, in the now famous stack diagram [27]:

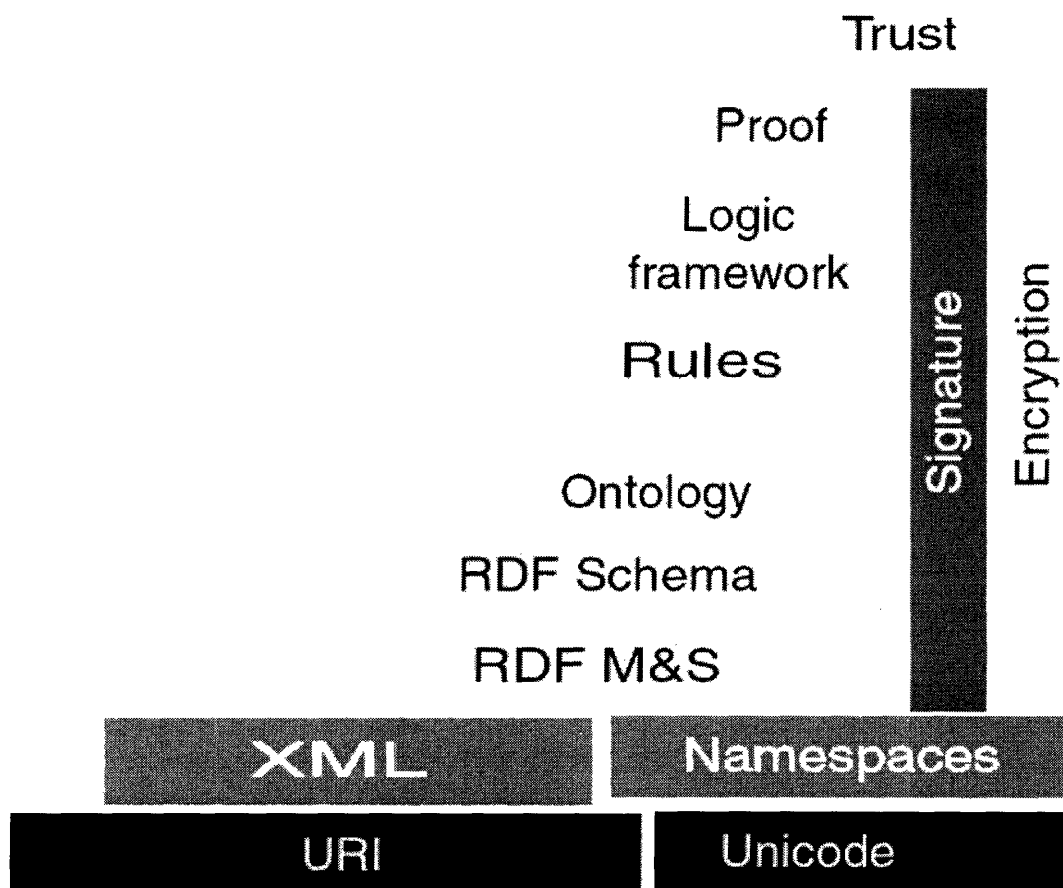


Figure 7: The Semantic Web Technologies Stack Diagram

Rules are the primary area of interest in this research. Specifically, we look at RuleML, an emerging standard for developing an open, interoperable, vendor-neutral XML/RDF based rule language. RuleML facilitates the exchange of

information between various systems, whether these are distributed software agents on the web or heterogeneous, corporate client-server applications.

There are *four* Currently Commercially Important (CCI) rule systems [19]

- Structured Query Language (SQL) – using views and trigger capabilities in databases. Trigger is a type of production rule, which is a specific database action invoked automatically and conditionally.
- Prolog – an AI language (Programming in Logic) oriented towards backward chaining rules. The meaning of a Prolog program can be explained very naturally using First Order Predicate Logic (FOPL)
- OPS5 consists of a set of production rules. Each rule has a precedent, or set of conditions that govern application of the rule, and an antecedent that define actions to take upon triggering the rule. OPS5 supports both forward-chaining and backward-chaining control models.
- ECA rules – The Event-Condition-Action model is followed in sentient (and by extension, reactive) applications. A predefined situation, typically a composite event pattern, is matched to trigger an action.

There are other ways of representing these rules in software applications, which are listed below [30].

- The most commonly used approach is by using if-then-else constructs in standard programming languages like C, C++, Java, etc.
- Knowledge Interchange Format (KIF): An emerging standard ANSI standard for representation of knowledge, including rules, between heterogeneous software systems like agents.

While the other methods are popular in e-commerce systems, KIF is not yet widely implemented in commercial applications. The following matrix is a mapping each of these approaches to industrial requirements:

Rule Systems Requirements	IF-THEN constructs In C/Java	Prolog	SQL Views	Database Triggers	Production Rules	KIF	RuleML (SCLP)
Interoperability through declarative semantics, ease of parsing rule sets	◐	◐	◐	◐	◐	●	●
Ability to handle non-monotonic rules : Negation as Failure, default rules	◑	◑	◑	◑	◑	○	●
Prioritized Conflict Handling implying modularity	◑	◑	◑	◑	◑	◐	●
Procedural Attachments for action invocation	●	●	●	●	●	◐	●
Ease of developing translators	◐	◑	◑	◑	◑	◐	●
Computational Tractability	◐	◐	◐	◐	◐	◐	●

Figure 8: Relative comparison of different CCI rule systems

3.2 Advantages of Semantic Rule-Based Trust Policy Management

There are several benefits of implementing business policies in a standardized semantic rule based language like RuleML. Some advantages are:

1. More interoperability, flexibility and reuse
2. Reduced system development, maintenance and training costs
3. Better, faster and cheaper security administration
4. Greater visibility into enterprise policy implementation leading to improved compliance
5. More resource sharing within the enterprise because of standardized access norms, leading to improved governance of corporate assets
6. Better conflict handling in policy-driven decisions through rich, expressive Trust Management language

Today, only a small part of the financial services industry is taking advantage of the expressive capabilities of rule engines to develop flexible applications. Yet, there is no single standard way of knowledge representation or communication between these rule systems. RuleML [19], (based on Situated Courteous Logic Programs), the emerging standard in rule-based knowledge representation offers users all the 6 benefits listed above, which proprietary solutions are unable to do today. With cooperation from vendors, RuleML will allow companies to leverage their existing BRE and BPM software while deciding on new investments.

3.3 Trends in the Business Rule Engine Market

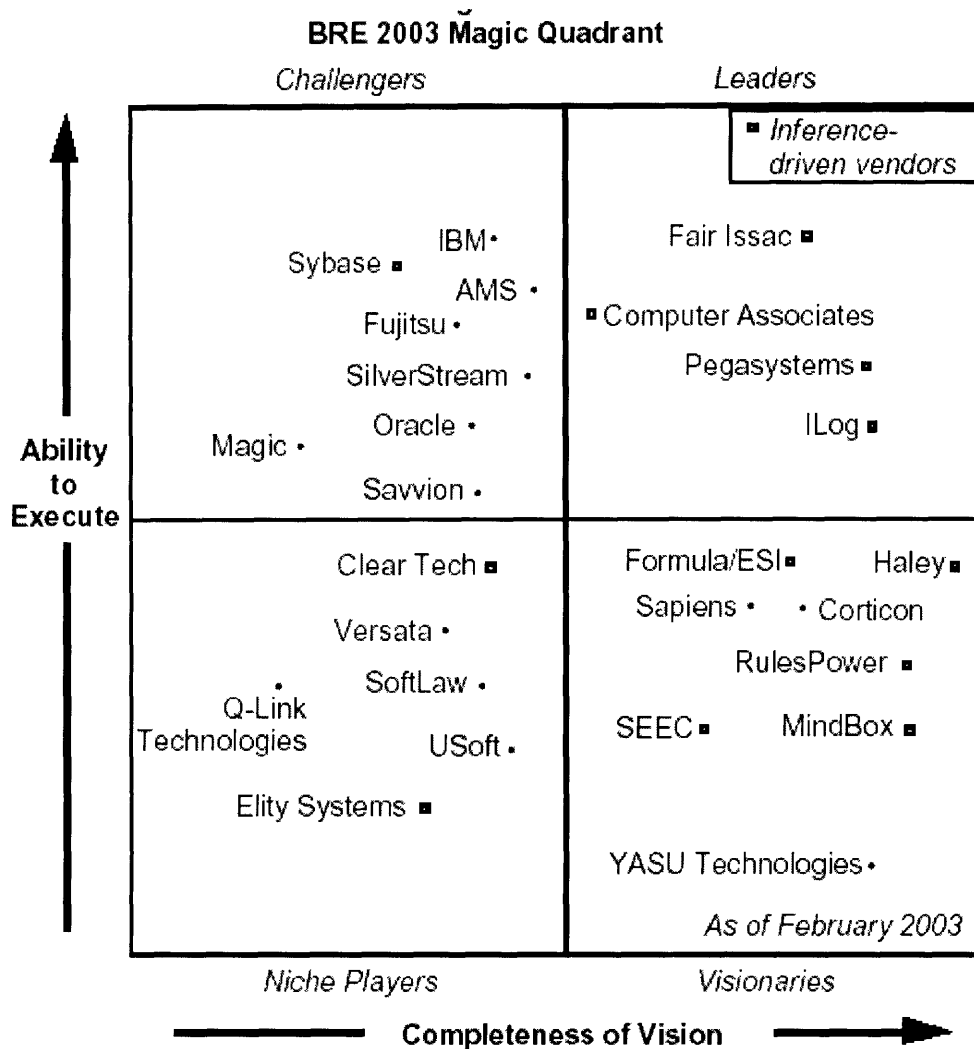
The Gartner Group finds that **Business Rule Engine (BRE)** market is picking up as firms strive to build agility in their systems, especially through flow control as typical in Business Process Management applications [50]. The size of the 2003 market was approximately 200M and growing. The Service Oriented Architecture (SOA) and the advent of Web Services will fuel the need for more rule engines. Another emerging market where BRE technology may play an interesting role is in the Business Activities Monitoring (BAM) industry. BAM is critical for infusing strategic agility, so that organizations can change swiftly by detecting a situation rather than changing reactively many weeks later. The following requirements in Figure 9 are often looked at in determining the efficacy of a rule engine. We also found that vendor skills and capabilities are widely dispersed, as evidenced in the Figure 10 magic quadrant diagram [50] below.

BRE Technical Requirements

<p>Advanced Inference</p> <ul style="list-style-type: none"> • Truth maintenance to support parallel rule execution • Inductive and deductive problem sets supported • Recursive rules supported • Rule taxonomies supported • Links to rule simulation capabilities • Agent or daemon links • Object inheritance supported • Multiple engine support 	<p>Versatility</p> <ul style="list-style-type: none"> • Easy to embed with and in other technologies • Multiple database management system support for rule repository • Links with legacy rule extraction vendors • Multiple rule methodologies support • Linked with world-class enterprise application integration vendors • Linked with world-class BPM vendors • Import/export/application programming interfaces (e.g., support for XML Metadata Interchange)
<p>Rule Management</p> <ul style="list-style-type: none"> • Rule extensibility • Rule mapping to owners and stewards • Rule change impact analysis purposes • Integration/coordination of distributed rules engines with a corporate "master" • Ability to rerun the engine for a point that has passed (e.g., after 1 January, able to rerun year-end jobs with 31 December rules) • Ability to enter new rules or changes to become effective on a future date (e.g., able to put in the rule changes for 1 January in December) • Rule consistency checks • Rule versioning • Release versioning and rollback • Rule security 	<p>Ease of Use</p> <ul style="list-style-type: none"> • Easy to change rules • Easy to test rules • Easy to visualize rule-firing sequences • Expert system/help • Rule-firing audit report capabilities • Rule views by project or role • Can be used as a wizard in development environments • Dynamic rule change supported • Rules separated from the engine • Constraints naturally supported <p>Performance</p> <ul style="list-style-type: none"> • High performance for large rule bases • Ability to share rule sets across multiple engines • Dynamic and static execution versions for performance • Multiple, cross-platform support

Source: Gartner Research (April 2003)

Figure 9 : Requirements of a strong rule engine 50]



Source: Gartner Research

Figure 10: Vendor dispersion for the BRE Market 50]

Business Process Management (BPM) has also become an important concept in the industry where rule engines play a key role. Till late nineties, BPM primarily meant document-centric workflow control. Then the Integrated Broker Suite (IBS) vendors led the way by positioning BPM as a system-to-system process management solution. Now, we can expect more IBS led development of human-

to-human and human to system BPM applications. The general definition of BPM is a solution that enables the design, execution, integration, monitoring and optimization of workflows among people and applications. BPM is deemed useful for processes that are frequent, costly, fragmented and compliance regulated. Here are some of the reasons that organizations take part in BPM activities [49]:

- a. Build better new processes faster
- b. Gain better visibility into current processes
- c. Avoid frictions during Mergers and Acquisitions
- d. Enable outsourcing of the “dull stuff”
- e. Buy software and implement packages better
- f. Control parallel processes by consolidating to core process
- g. Identify more opportunities where easy work can be automated
- h. Partner selection and creation opportunities
- i. Do things better with optimized processes
- j. Stay ahead of compliance requirements through process model analysis.
- k. Stay hungry – move faster for agility and policy management

The following chart (Figure 11) shows the current distribution of BPM vendors along the dimensions of niche players versus general-purpose vendors [48]:

The 2Q03 Pure-Play BPM Magic Quadrant

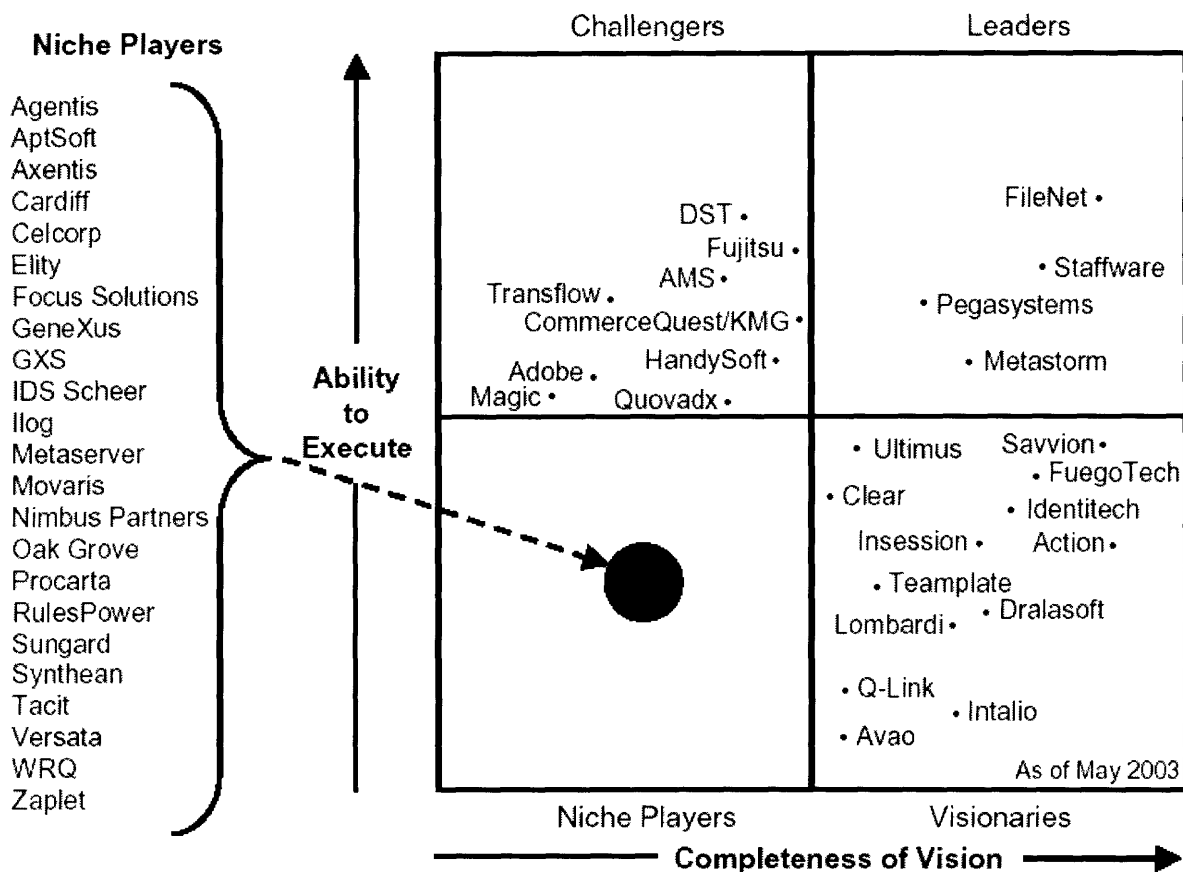


Figure 11: The popular BPM market and the players 50[48]

Over time, the Business Process Management (BPM) vendors are expected to develop or buy their own “good enough” rule engines, and hence the pure BRE market may disappear [50]. However, businesses with inference engines or multiple engine types have a greater chance of survival since they are good for solving complex processing as well as simple applications.

4. Chapter 4

4.1 Advantages and Capabilities of Semantic Web Rules Technology for Trust in Financial Services Industry

In Chapter 2, we reviewed the major business challenges in the financial services industry. Decentralized Trust Management [2] is a central theme in all these issues, from Check 21 to GLB implementations to Sarbanes Oxley (SOX) compliance. A common characteristic of such business applications is that they are all rule-driven and policy-dependent. Thus technologies that handle rule processing well should be desirable in developing software solutions within financial services.

In Chapter 3, we talked about semantic web rules and RuleML, an emerging standard for knowledge representation of business rules. A completed product base for RuleML does not exist at this point but since it is based on Situated Courteous logic Programs (SCLP), we have used a reference implementation called IBM Commonrules to represent the capabilities of RuleML. We discussed the general advantages of semantic web rules for use as a policy language including greater interoperability, cost reduction and better conflict handling.

In this section, we will analyze the business requirements of some of the challenges from Chapter 2, discuss technology solutions and determine whether semantic rules can be used to solve some of these problems effectively.

4.2 Consumer Householding

For example, to make **Consumer Householding** possible, we need to determine how to relate certain account holders of different services within a financial company. A father may have a high net worth retirement and checking account jointly with his wife, one adult son may have a separate premium brokerage account while the younger under-18 daughter may be enrolled in a trustee education account. Quality data is critical to capturing the complete customer view. In a perfect world, if each customer willingly provided true information about other accountholder relatives, it would be a simple matter of cross-referencing each portfolio – however, such an ideal situation is unlikely. Hence, we need to combine intelligence and business data to make relational deductions. Some examples of rules that may help us to make this determination are:

- (a) If the street address of all accountholders match, it is likely they are related.
- (b) If the full name of an accountholder matches one of the names on a joint account, and the first accountholder is below 30 years in age which the second is greater than 50, it is possible the former could be a child of the latter.
- (c) If the SSN of the beneficiary in any trustee or retirement account matches that on any other primary account, it is quite possible that the two accounts belong to members of the same family.

Clearly, large-scale database technologies and remote data acquisition capabilities are required to store snapshots of the Households. The process to generate household entries in a database is very rule-dependent. Interoperability is a key requirement here, especially since portfolio of different accountholders may exist in different parts of the organization, possibly in separate business units. To institute Householding, parts of the business must determine common ways of accessing data and understanding the meaning of data elements. Secondly, we would need flexibility because a rule or data element change in any one application will affect Householding decisions. Third, once we establish Householding relationships successfully, we need further rules to segment the universe and make the data useful. Marketing, for example, may want to offer special service incentives to people falling within a certain demographics. These offers may change the following week. Fourth, we need a simple, yet rich language to express both forward chaining (e.g. if rich household, offer Portfolio Advisory Services) and inference logic (if last names on two accounts match and addresses match, clients must be related). Finally, there are some rules that may contrast each other. Two customers A and B may be listed as parents of C, but A's marital status is divorced; hence A and B may not be considered in the same household. Thus we will need a mechanism to specify prioritized conflict handling or override mechanism.

Per interviews with leading financial companies, we have seen that today, 99% of all their business rules are hard-coded in applications or represented through some combination of procedural programs and SQL databases. These conditions pose two major problems. First, the very discovery of policies can be a major challenge, as they require people from each business unit with specific knowledge of how the rules work to communicate together. Secondly, it would be tremendously tedious to build and maintain the household database if all the IT systems were developed in different technologies. Third, the household database itself would have been a silo application where maintenance would have been tricky once the original developers left. It also would have reinvented the wheel for many rule management, execution and prioritization logic.

If we had used commercial rule engines like those from ILOG, Pega and Fair Isaac, the third problem would have been mitigated, though not completely eliminated (Ref. Chapter 3). Going with a single vendor solution across the entire enterprise would also have reduced the pains of integration and visibility issues. However, this may be a disastrous strategy for the customer in terms of allowing vendor lock-in. Several issues can happen over time:

- (a) Vendor risk - The vendor may stop supporting the installed product version, insisting that the customer move to a more expensive version. It can also go out of business. If the product is deeply entrenched in the customer environment, the client may be essentially held hostage by the vendor.

- (b) Technology risk – Most software methodologies are only useful for a few years, before a disruptive technology sweeps in and offers much greater value. If a vendor is pervasive across the enterprise, the switching costs are too high, both in terms of price paid and time expended. A leading financial services company in Boston bought a major card-services workflow product once, which later wasn't deemed useful and they had to replace it. It took them four years to do so!
- (c) Growth risk – If the firm merges with another organization, the dominant standards may change instantly. The benefits of internal product platform standardization will be lost and will have to be changed significantly, leading to similar problems as in procedural programs.

Thus, upon evaluating all such options, it seems evident that a standardized rule technology will be the best approach in resolving a difficult problem like Householding. As a powerful standard, SCLP featured RuleML meets all the challenges described in this example. It allows ease of integration between standardized system, a power expressive rule language with declarative semantics, polynomial-level tractability, procedural attachments and prioritized conflict handling. The rule language is simple and easy to maintain across the universe, obviating the need for analysts with deep system knowledge to demystify the business logic. In a standardized environment, the usability of the application from a pure rules perspective does not cease when M&A activities

occur since its relatively easy to understand the policies of the new system and incorporate them in the Householding application. The challenges of data acquisition from the new system may still be valid, but since that is a batch process, one can assume that any commercial database will support data export operations.

One objection that critiques have raised to rule engines is because of perceived performance latency. However the batch mode of Householding application overcomes this objection since real-time responses will not be required.

4.3 Compliance Regulations

The approach to meeting regulatory challenges, as with SOX, HIPAA, GLB etc combines business process changes with IT system upgrades. The key insight here [44] is not to implement a point solution for each legislation but rather look at the vulnerability gaps within an organization that most regulations seek to bridge. A longer-term strategy for compliance management is often IT dependent, especially for better BPM, accounting, audit, policy awareness and security. Over time, it is conceivable that corporations will improvise an enterprise system for streamlining compliance across the firms that address the core functional issues listed above. The outstanding challenge then will be to implement new or

changing policies in the existing system as quickly and easily as possible, reducing the enormous burden of regulatory compliance.

If we can assume that most major regulations target key exposure areas like accounting, reporting, audit etc, then all that changes with new laws are business rules. Hence, we must have a mechanism to implement new rules into the IT process. These updated regulations may override older working principles. Our goal is to establish as small an application footprint as possible to avoid building a humongous repository of interdependent rules that will require a large team to maintain. Even a greater risk is low management visibility into the effective enforcement of rules. If rule systems are long and difficult to understand even at a technical level, senior management can have no confidence that compliance regulations are being met. In case of Sarbanes Oxley, this exposure represents a noteworthy prosecution threat for the CxO level individuals, who may face penalties from steep fines to potential time in jail, as has happened with Enron and Tyco management.

To reduce involuntary errors from policy obfuscation, the key requirement is for a flexible rule based system that can succinctly incorporate the changing needs of the business as well as of the regulatory environment. RuleML scores high on this list especially because of its strong expressive power, prioritized conflict management between multiple rules and capabilities for procedural attachments.

This last feature (procedural attachments) reduces cost of software development by providing exit gateways to fetch information at runtime from a different source. For example, in a backward chaining logic to test whether a bank account should be suspected for use in terrorism (per Patriot Act), the following details may be necessary:

- (a) If the accountholder tries to send (wire or check) any amount over \$1000 to a “blacklisted” destination country or suspected terrorist organization, the bank should report it.
- (b) Any sudden withdrawal of a sum greater than \$20000 should be reported unless the accountholder has consistently done so in the past.

Either of the two rules are sufficient conditions for the bank to file a SAR (suspicious activity report). It may be that the first condition holds true under many circumstances, assuming that certain accountholders send money to their families in these countries for legitimate reasons, though the bank has to report it. Thus we could use an in-memory procedural attachment to retrieve the current collection of blacklisted nations from a business table. Thus, unlike forward chaining, this mechanism provides flexibility of not needing all input data upfront but rather going out and getting the required data as and when desired. The effectiveness of semantic web rules is quite clear in this example.

4.4 Security Authorization and XACML

A major debate continues around best solutions for **Identity** Management and Authorization. Standards like SAML and **XACML** are evolving to minimize vendor-specific impact to the firm, reduce the total cost of ownership, enable easier changes in security policies and demonstrate a “best practices” implementation commitment towards protecting enterprise information assets. Public Key and Role Based Access Control (RBAC) Technologies are useful here, but they do not provide a complete solution in an heterogeneous environment. As we have seen before, RBAC is limited delivering access control based on static groups. Some vendors specify additional policy languages for fine-grained authorization decisions, but these are cumbersome, less understood and hence little used. In addition, security is the most stovepiped technology, since every deadline-driven application built in financial companies tends to come wrapped with its unique security architecture. One suggested architecture to resolve this issue comes from the OASIS TC on XACML as described below.

XACML specifications contain a policy and rule section that does provide the capability of applying rules for specifying authorization decisions. Rules [40] are always defined in the context of a policy and consist of the following components:

- A target – refers to a set of resources, subject, action and environment to which the rule is supposed to apply.

- An effect – the rule writer’s intended consequence for a True evaluation for the rule. The allowed values are Permit and Deny.
- A condition – represents a Boolean expression that refines the applicability of the rule beyond the predicates implied by its target. Rules are not exchanged directly between system entities, but combined into policies.

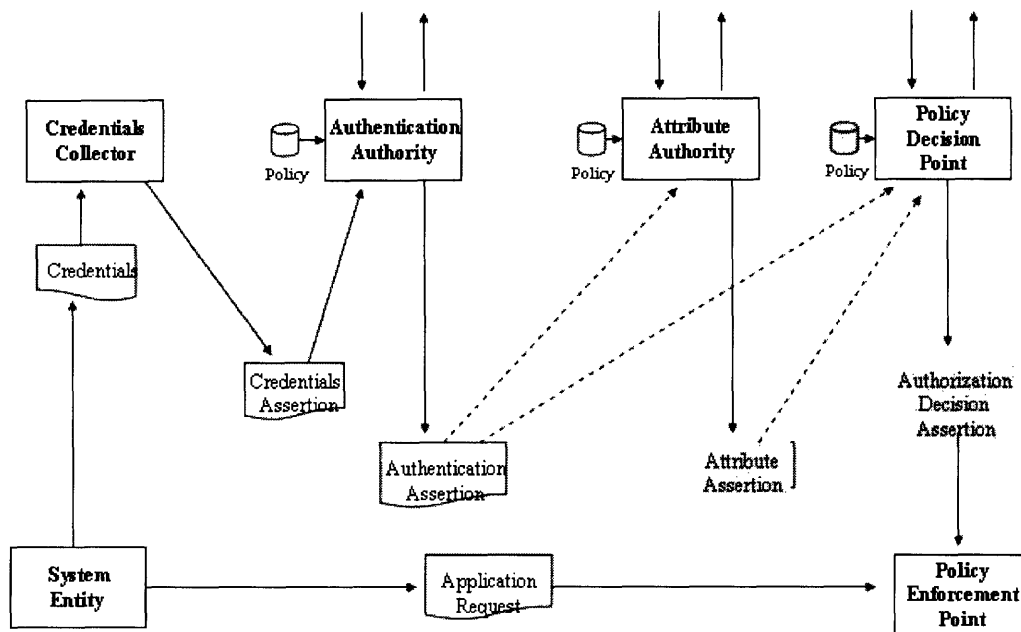


Figure 13: The Authorization Model for XACML 50[40]

In some ways, this is similar to the ECA (Event-Condition-Action) model. XACML adopts a relatively minimalist approach towards Knowledge Representation, and backward chaining is not supported. In addition, prioritized conflict handling and

expressiveness of the language are rather limited in comparison with RuleML. On the other hand, XACML has a singular design that makes it attractive from an authorization standpoint. For example, its policy evaluation results in binary decisions - permit or deny access control, though it also supports exception handling and indeterminate state (catch all condition). It has a number of useful built-in functions that allow easy computations and string manipulations.

In XACML, the Policy Decision Point (PDP) has the key role of evaluating different rules. The standard was created with traditional rule engine capabilities in mind. However, commercial grade rule engines developed over many years can handle much more complex operations than the PDP. Here is where integration capabilities with RuleML can help. RuleML compatibility can allow XACML to expose authorization services to other rule-based business applications. In a greater sense, RuleML (with SCLP) can be the foundation for a practical implementation of XACML. The table below shows how the different policy language requirements of XACML [40] map to capabilities in RuleML.

XACML Requirement	Supported By RuleML	Relationship Description
Capability to aggregate rules into a single policy that applies to a specific decision request e.g. Policyset > Policy > Rules	Yes	RuleML rulebase construct allows import and export operations. Rule units can be combined under one "tagged" rule in Courteous Logic Programs (CLP). Using AND, OR, MUTEX constructs, smaller rules can be integrated into a larger "tagged" rule.
To allow definition of a flexible procedure by which rules and policies are combined	Yes	This capability is automatically achieved in CLP. The CLP semantics, implied in the Commonrules rule evaluation mechanism looks at all applicable rules before reaching a decision. Multiple overrides are supported and

		"conditional override" feature allows more flexibility in decision making than XACML
To provide a method for dealing with multiple subjects	Yes	SCLP can meet various degrees of this requirement. We can create a separate rule for each subject. Sensors can separately retrieve data for each subject category while effectors can deliver differentiated authorization response.
Make authorization decisions based on subject/resource attributes	Yes	SCLP does not specify any "out of the box" attribute for policy attributes but provides complete freedom of unlimited attribute definition, plus both long term and short term facts. Again, sensors can be used to collect data from external sources, which are a superset of what XACML needs, where most attribute values are passed in. In addition, RuleML can also utilize information defined in further rulesets, or in OWL (new W3 standard for Ontologies) or RDFs.
Handle multi-value attributes	Yes	Though there is no direct support for multi-valued string attributes, java objects, Practical rule languages and RDFs are supported, each of which can encapsulate arrays of attributes.
Ability to authorize based on content of an information resource	Yes	RuleML has support for procedural attachments, which can get the data value from even non-XML documents. A set of built-in comparison functions allows each rule representation based on resource values. It can also access policies as data very simply by import or URI references.
Provide mathematical and logical operators on all attributes	Yes	RuleML has generic extensibility capability and support for multiple practical rule languages (e.g. SWRL which has many built-in functional constructs) Most standard mathematical and logical operators are supported in CommonRules inference engine, including object types.
Ability to handle distributed policy components while abstracting the way to locate or retrieve them.	Yes	Rules need to be created to link specific policies to resources. For other procedures, sensors can be created along with Java classes to retrieve policy elements for target resources from databases.
Rapidly identify the applicable policy based on attribute values	Yes	The prioritized conflict handling capability of RuleML (SCLP) would achieve this objective easily. Rule inferencing can be performed fast. As before, database indexing capabilities can be used to speed up performance in conjunction with the rule engine.
Provide abstraction layer insulating policy writer from details of application environment	Yes	The XML or RDF interlingua translating capacity for converting rules to/from other non-XML or XML/RDF rule systems, KIF, Jess or XSB allow great flexibility for abstraction. This has been demonstrated in the SweetRules [18] and SweetJess [20] implementation.
Mechanism to specify actions accompanying policy enforcement	Yes	This requirement can be met completely through use of effector functions following consequences.

Figure 14: Mapping XACML requirements to RuleML

Thus, we can conclude that almost all requirements of XACML are directly met by existing features of RuleML or can be derived from supported capabilities like procedural attachments or prioritized conflict handling.

4.5 Fast Moving Markets

As a final discussion of how rule systems play a critical role in business performance, let us consider fast-moving markets within the financial industry. One example is program trading of equities and the related circuit breaker laws mandated on exchanges by the SEC. Program trading allows large institutional clients to use mathematical models and software capabilities to execute automated buy and sell orders rapidly milliseconds ahead of the market. Price is the only factor considered by program traders, ignoring intrinsic value of the security. Almost 30% of NYSE daily volume comes from program trading. If markets are falling or rising rapidly, continued program trading can generate large positive feedback loops, endangering the liquidity of the affected securities.

SEC introduced a circuit breaker rule [46] to program trading, following Black Monday in 1987 which led to a largest percentage point decline of the Dow in its history. Under this regulation, cross-market trading is briefly halted every time the Dow falls by 10%, 20% or 30% of the value of the exchange, as set in the prior quarter. In addition, collar rules are triggered once the Dow moves by 2% of prior

closing value. Program traders are required to run executions in the direction that will stabilize the stock price. Similarly, price limits kick in dynamically for future contracts during price declines. Such limits are automatically removed at 3-30 PM each day or 10 minutes after the limit thresholds are reached.

Program trading policies can be clearly represented in RuleML. These are examples where threshold values change frequently and the policies can be updated any time by the governing agencies, based on market situation. We need the flexibility to rapidly specify a new rule with override conditions and have confidence that the system will take care of conflict resolution and supremacy concerns within rulesets [46]. Procedural attachments are important since they allow the extraction of current trading limits and applicable policies.

Another dynamic industry where extremely rapid response is required to changing conditions is email spam and network vulnerabilities market. Financial Services industry is often the target of major viruses, as depicted in the following diagram, especially with the advent of “zero day” blended threats [52].

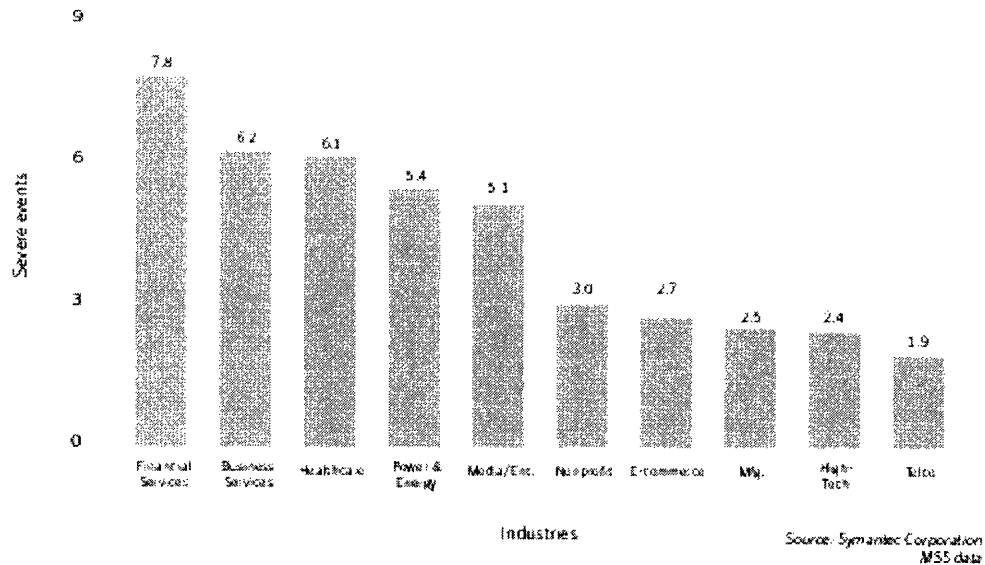


Figure 15: Several vulnerability threats across different industries [52]

Too much spam has compelled network administrators to establish policies in their systems to keep unwanted mail from reaching the targets. Anti-spam techniques often include mail-filtering mechanisms, where mail from certain domain sources or containing specific textual phrases are blocked. The problem here is that there is limited intelligence, compared to that of human beings, to decide whether a message is valid or should be discarded. Often, it depends on what the end user considers as spam. For example, a web site may be sending standard newsletters that need to be ignored, but when a customer finally makes a purchase from that online company, he wants to get the product information delivered over email. Normal email blocks cannot take care of this situation. But SCLP RuleML can be employed to build rules for allowing non-spam messages to

get through. Hence this is another example of how this technology can find widespread application.

Throughout this chapter, we have examined real world examples of business challenges in the financial services industry and the emerging technologies available to combat such issues. In almost every case, from Consumer Householding to Check 21 to Enterprise Security, we have seen that Semantic Web rules provide a powerful mechanism for knowledge representation, essential for meeting the granular requirements of each business need. We have also argued in favor of adopting standardized platforms like RuleML in developing knowledge based systems, both from the customer and the vendor perspective. In the next chapter, my goal is to demonstrate the capabilities and expressive power of RuleML in solving some business problems in finance industry. Further research will certainly unravel more examples of RuleML's universal benefits.

5. Chapter 5

5.1 Examples of Business Rule based Authorization in the financial industry

Business rules are inherent in different parts of the financial industry. Here are some common examples:

Classification	Application	Rule
Mutual Funds	Rep trading	<i>Blue Sky</i> : State restrictions for rep's customers
Mortgage Company	Credit Application	TRW upon receiving credit application must have a way of securely identifying the request.
Brokerage	Margin trading	Must compute current balances and margin rules before allowing trade.
Insurance	File Claims	Policy States and Policy type must match for claims to be processed.
Bank	Online Banking	Every user can look at own account
All	House holding	A Business rule to provide one aggregation of accounts of both customer and his family in this silos.

Figure 16: Examples of business rule usage in financial services

5.2 Case Implementations using Semantic Web Rules

In this section, we have attempted to provide example implementations of some real cases requiring rule-based processing from the financial services industry.

Many of these examples emerged during discussions with key interviewees from the financial services, vendors and academicians. We have focused on examples which demonstrate the various capabilities of semantic rule systems, like expressiveness, prioritized conflict handling, modularity etc.

Most of the SCLP example scenarios below are self-explanatory. From a syntax perspective [17, 16], CNEG stands for classical negation, such that {CNEG p} implies “I believe that p is false”, whereas {FNEG p} means, “I don’t believe p is true”. The overrides construct demonstrates prioritized conflict handling capabilities of one rule over another.

5.2.1 Case I: Credit Card Verification System for Electronic Transactions

Description: Electronic purchases of goods and services on the web are often paid for by credit cards. Because of rising identity thefts leading to greater incidence of frauds on the net, credit card companies have tightened the security requirements in online transactions. It used to be that if a customer’s credit card account was in good standing and not expired, the transaction was successful. Over the past few years, several new verification rules have been added which include ability to perform billing address verification (AVS) and to match card verification code (CVC), present on the back of the physical card. In addition, the merchant often maintains a separate database of “blacklisted” customers, using which a service provider can deny a transaction even if the has passed all the

prior tests. The AVS test can also be used to override a card account in good standing.

This SCLP program checks creditworthiness, largely the validity of a credit card for an electronic transaction using a set of governing rules. Three independent parties create these rules:

(a) The issuing bank

(b) The "merchant", which is an electronic service provider (e.g. a web site); in our example it's Amazon.com.

(c) A 3rd party fraud alert reporting service (which can be viewed as an agent)

This rulebase specifies the policies of the merchant. However, the merchant gets much of its rules and facts from the bank and the fraud reporting agent.

Policies created by the bank include that:

- The card must be in good standing, must not be expired or above limit. In addition, from a security standpoint,
- The purchaser's credit card billing street address must match the cardholder's billing address on record (AVS); and
- The purchaser's Card Verification Code (CVC) must equal that of the cardholder's CVC on record.

If any of these conditions are false, the transaction is rejected. If all these conditions are true, the transaction is good from the bank's perspective.

The fraud alert reporting service maintains a database of people and reports whether a given card/purchaser has a fraudulent transaction history. Reports from the fraud alert reporting service have high priority -- they take precedence over any authorization from the other rules -- i.e. those provided by the bank. Below, Fraudscreen.net is an example such fraud screening party.

In addition, the merchant service provider maintains its own database, listing customers with whom it has had trouble dealing in the past. Thus the service provider itself has a policy to disallow certain further transactions.

```
/*
SCLP Rulebase in IBM CommonRules SCLPfile format: creditcheck
Version: 2
Date: March 9, 2004
Authors:
- Chitro Neogy, MOT 2004, Sloan School of Management chitro@sloan.mit.edu>
- Benjamin Grossof <bgrosof@mit.edu> http://ebusiness.mit.edu/bgrosof
*/

/* the following group of rules are created by the bank/credit-card-company,
then adopted/imported as a group/module by the merchant. */

/* bank says by default the transaction is authorized if the card is in
good standing */
<bankResp>
  if checkTran(?Requester)
  then
    transactionValid(self,?Requester);

/* bank says the transaction is disallowed if the card is expired. */
<cardRules1>
  if checkCardDet(?Requester, ?accountLimit, ?exp_flag, ?cardholderAddr, ?cardholderCVC)
and
  checkTranDet(?Requester, ?tranAddr, ?tranCVC) and
  notEquals(?exp_flag, "false")
  then
    CNEG transactionValid(self,?Requester);

/* bank says the transaction is disallowed if the Card Verification Code
does not match what's on file for the card. */
```

```

<cardRules2>
    if
        checkCardDet(?Requester, ?accountLimit, ?exp_flag, ?cardholderAddr, ?cardholderCVC) and
        checkTranDet(?Requester, ?tranAddr, ?tranCVC) and
        notEquals(?tranCVC, ?cardholderCVC)
    then
        CNEG transactionValid(self,?Requester);

/* bank says the transaction is disallowed if the card is above
   its account limit. */
<cardRules3>
    if
        checkCardDet(?Requester, ?accountLimit, ?exp_flag, ?cardholderAddr, ?cardholderCVC) and
        checkTranDet(?Requester, ?tranAddr, ?tranCVC) and
        lessThan(?accountLimit, 25)
    then
        CNEG transactionValid(self,?Requester);

/* bank says the transaction is disallowed if cardholder address does not
   match what's on file for the card. */
<cardRules4>
    if
        checkCardDet(?Requester, ?accountLimit, ?exp_flag, ?cardholderAddr, ?cardholderCVC) and
        checkTranDet(?Requester, ?tranAddr, ?tranCVC) and
        notEquals(?tranAddr,?cardholderAddr)
    then
        CNEG transactionValid(self,?Requester);

/* The following rules are additional policies of the merchant.
   */

<fraudResp>
    if
        cardGood(?FraudFirm,?Requester,bad) and
        fraudExpert(recommenderService,?FraudFirm)
    then
        CNEG transactionValid(self,?Requester);

<svcProviderResp>
    if customerRating(?svcProv, ?Requester, bad)
    then
        CNEG transactionValid(self,?Requester);

overrides(cardRules1, bankResp);
overrides(cardRules2, bankResp);
overrides(cardRules3, bankResp);
overrides(cardRules4, bankResp);
overrides(fraudResp, svcProviderResp);
overrides(svcProviderResp, bankResp);
overrides(fraudResp, bankResp);
fraudExpert(recommenderService,Fraudscreen.net);

```

```
/* The following groups of facts each specify a
particular case scenario of a (requested) transaction.
*/
```

```
/* Joe has a card in good standing, unexpired, below the account limit,
and his address matches. Plus the fraud alert service rates him fine,
and the merchant customer rating is fine too.
```

```
  The policies will thus imply that his transaction ought to be
  authorized. */
```

```
checkTran(Joe);
checkCardDet(Joe, 50, "false", 13, 702);
checkTranDet(Joe, 13, 702);
cardGood(Fraudscreen.net,Joe,good);
customerRating(Amazon.com, Joe, good);
```

```
/* Mary has a card in good standing, below the account limit, her
address matches, and her fraud report and customer rating are good.
But her card is expired, and the CVC does not match.
```

```
  Thus the policies will imply that her transaction ought to be disallowed.
```

```
*/
checkTran(Mary);
checkCardDet(Mary, 290, "true", 27, 546);
checkTranDet(Mary, 27, 545);
cardGood(Fraudscreen.net,Mary,good);
customerRating(Amazon.com, Mary, good);
```

```
/* Andy has a card in good standing, below the account limit, his address
and CVC match, and his fraud report is good. But his card is expired and
his customer rating is bad.
```

```
  Thus the policies will imply that his transaction ought to be disallowed.
```

```
*/
checkTran(Andy);
checkCardDet(Andy, 30, "true", 14, 703);
checkTranDet(Andy, 14, 703);
cardGood(Fraudscreen.net,Andy,good);
customerRating(Amazon.com, Andy, bad);
```

We obtained the following results from the system, which does match what is expected from the system:

```
SCLPEngine: Adorned Derived Conclusions:
CNEG transactionValid_c_3(self, Mary);
transactionValid_c_2(self, Joe);
transactionValid_c_2(self, Mary);
transactionValid_r_2(self, Mary);
transactionValid_u(self, Joe);
CNEG transactionValid_u(self, Mary);
transactionValid(self, Joe);
CNEG transactionValid(self, Mary);
```

5.2.2 Case II: Check Clearing for the 21st Century Act

Check 21 is hyped as the biggest driver in the banking industry. As mentioned before, it is a regulation which will change the way deposited checks are handled in this country by creating the notion of digital substitute checks, based on a set of criteria identified by the Federal Reserve Board. In this example, we will develop a rule model for managing trust in substitute checks.

Very few organizations have actually created a rulebase for Check21 at this point since part of the regulation is yet to be finalized. The following representation considers the major tenets of this new law and should be considered a work in progress. This program checks whether check being processed in a post-check21 timeframe is valid or not. The governing rules here are based on new Federal Reserve Board security instrument called the substitute check, a.k.a. the Image Replacement Document (IRD).

Check 21 Act, which goes into effect on Oct, 2004 will require banking institutions to accept a substitute check in lieu of an original check provided it meets all of the rules specified by the X9 committee (www.x9.org), and specifically as laid out in DSTU X9.90 document, which describes substitute checks. Thus, these rules are created by at least three independent parties:

- (a) Federal Reserve Board
- (b) The ANSI X9 committee

(c) The individual banks, especially when it comes to the question of accepting electronic images of checks for forward collection or return. Note that while Check21 facilitates exchanges of electronic check copies, it does not require it, so banks must mutually agree on the standards and validity of electronic images.

X9 organization provides specifications of valid substitute checks. Individual banks decide whether to accept electronic images of substitute checks or not to facilitate Straight-Through Check Processing (STCP), a big drive in the industry to reduce costs.

Our Model in SCLP

```
/*
SCLP Rulebase in IBM CommonRules SCLPfile format: check21
Version: 1
Date: April 2, 2004
Authors:
- Chitro Neogy, MOT 2004, Sloan School of Management chitro@sloan.mit.edu>
*/
```

```
/* the following rules are created by the Federal Reserve Board */
```

```
/* If check is original or is a substitute check which meets all X9 criteria, bears the front and back copy
of the original check, had a valid MICR, contains information of BOFD, then check is good
In this section, checkType can be original or substitute
MICR and Bank of First Deposit must be valid
Note that a MICR unique identifies a check.
*/
<FRBRules>
  if FRBRulesOK(?checkType, ?frontImageMatch, ?backImageMatch, ?MICR, ?BOFDValid, bad)
  then
    CNEG checkValid(self,?MICR);
```

```
/* Next we have specifications by X9 committee */
```

```
/*
X9 standards group have several size and content related specifications on substitute checks. These
are:
*/
```

```
<X9rules>
    if X9RulesOK(?MICR, ?sizeOK) and
    notEquals(?sizeOK, "yes")
    then
    CNEG checkValid(self, ?MICR);
```

```
/* Finally, we have bank specific rules. At this point, this includes whether banks want to have
electronic image exchange in processing of substitute checks
*/
```

```
<Bankrules>
    if
    checkIsElectronic(?MICR) and
    BanksAcceptEChecks(?bank1, ?bank2)
    then
    checkValid(self, ?MICR);
```

```
/** Specific country rules exist - if country of origin is not US, substitute checks can'tbe used
Note: Substitute checks are identified by EPC code (position 44) which must be 2 or 5**/
```

```
<Countryrules1>
    if origin(?countryOrigin, ?MICR, ?EPC, good) and
    equals(?EPC, 5) and
    notEquals(?countryOrigin, "US")
    then CNEG checkValid(self, ?MICR);
```

```
<Countryrules2>
    if origin(?countryOrigin, ?MICR, ?EPC, good) and
    equals(?EPC, 2) and
    notEquals(?countryOrigin, "US")
    then CNEG checkValid(self, ?MICR);
```

```
overrides(Countryrules1, X9rules);
overrides(Countryrules2, X9rules);
overrides(X9rules, FRBRules);
overrides(FRBRules, Bankrules);
```

```
/* The following groups of facts each specify a
particular case scenario of a (requested) transaction.
*/
```

```
/*
Fleet Bank and HSBC has an electronic check exchange agreement. In this example, all the
requirements from FRB, including front and back image, MICR, BOFD information is good. The EPC
code is valid for this MICR. However HSBC is a foreign bank and hence its checks cannot be
substituted*/
```

```
checkIsElectronic("000067894");
BanksAcceptEChecks(Fleet, HSBC);
FRBRulesOK(substitute, good, good, "000067894", good, good);
origin("HK", "000067894", 2, good);
X9RulesOK("000067894",yes);
```

```
/*
```

Fleet Bank and Wachovia has an electronic check exchange agreement. In this example, all the requirements from FRB, including front and back image, MICR, BOFD information is good. However X9Rules are not met in terms of the EPC code */

```
checkIsElectronic("000067895");  
BanksAcceptEChecks(Fleet, Wachovia);  
FRBRulesOK(substitute, good, good, "000067895", good, good);  
origin("US", "000067895", 3, good);  
X9RulesOK("000067895",yes);
```

/*

Fleet Bank and Citibank has an electronic check exchange agreement. In this example, all the requirements from FRB, including front and back image, MICR, BOFD information is good. However X9Rules are not met in terms of size */

```
checkIsElectronic("000067896");  
BanksAcceptEChecks(Fleet, Citibank);  
FRBRulesOK(substitute, good, good, "000067896", good, good);  
origin("US", "000067896", 3, good);  
X9RulesOK("000067896",no);
```

SCLPEngine: Adorned Derived Conclusions:

```
CNEG checkValid_c_5(self, "000067894");  
checkValid_c_3(self, "000067894");  
checkValid_c_3(self, "000067895");  
checkValid_c_3(self, "000067896");  
CNEG checkValid_c_2(self, "000067894");  
CNEG checkValid_c_2(self, "000067895");  
CNEG checkValid_c_2(self, "000067896");  
checkValid_u(self, "000067894");  
checkValid_u(self, "000067895");  
checkValid_u(self, "000067896");  
CNEG checkValid_u(self, "000067894");  
CNEG checkValid_u(self, "000067895");  
CNEG checkValid_u(self, "000067896");  
CNEG checkValid_s(self, "000067894");  
CNEG checkValid_s(self, "000067895");  
CNEG checkValid_s(self, "000067896");  
checkValid_s(self, "000067894");  
checkValid_s(self, "000067895");  
checkValid_s(self, "000067896");
```

SCLPEngine: Processing ends.....

5.2.3 Case III: Brokerage System Account Access

Trading systems perform millions of transactions daily, where security for individual customer accounts is extremely high. This is a very competitive industry

and companies are very protective of their customer assets. Many high net-worth portfolios are serviced by registered reps that will not share account information even with other broker-dealers in their same company. Hence a computerized trading system must be able to build walls between customer domains of each such rep and firm. The example given below is an implementation of a real system of a leading financial services firm. Two types of companies are served by this system – retail and correspondent. Different access rules apply for every type of user and account which are explained in the comments of the SCLP example below, which checks customer account access permissions by a registered representative of a brokerage firm. This logic mimics the production implementation of a leading firm. This company has millions of accounts that are considered either retail or correspondent. The retail world is simple because all customers belong to the brokerage organization and we'll assume that any rep of that firm with system access, who is also an employee of the retail division, can access the account.

The correspondent world is very complex. A correspondent is a broker/dealer, financial planner or bank providing a number of services to their own clients, but do not own a high-performing trading system. Hence they use our brokerage firm's proven computer system to service customers, which paying a fee to the brokerage firm for system use. Thus one single software system (typically a mainframe) holds man end customer accounts, some of which are retail and the

rest belongs to correspondents. There are thousands of correspondent firms, ranging from 1-2 person Financial Planning Houses to national banks with multiple branches. While the retail business growth was slow, correspondent business was booming since very few companies had resources to build a similar grade trading system.

The following rules were needed to look at a customer account on the correspondent side:

(a) The accessor's login profile was used to determine his company and ensure that the account belonged to the same company.

(b) Some correspondents had several branches, organized regionally. Every brokerage login provided information about a rep's branch code. The first three digits of the account number is the branch code (e.g. a/c 301123456 has 301 as branch code). If the rep's branch matches the account's branch, he is normally eligible to access account.

(c) In addition, some branches (typically at banks) have jurisdiction over other branches within a certain correspondent bank. This is a relatively arbitrary mapping (e.g. 301 can look at all accounts of 301, 302, 304) provided by the bank and is maintained in a central table. This method bypasses the restrictions in (b), so if I'm in branch 301, I can access an account 304123456 as well.

(d) We're not over yet! Some correspondents have the option to turn on a special permission called registered rep ownership at the company level. If this is on, that

means individual reps within a single branch are competing for customers and hence do not want their colleagues to see their customer accounts. In such cases, there are two fields on the customer account called the RR1 and RR2 fields, which stored the login id of two different representatives. The accessing rep's login must match the value in either of these two fields to establish ownership. There are even more rules on the mutual fund side, but the current complexity level is sufficient for demonstration purposes. This is a good example of "Business Authorization" described earlier since the authorization decision depends on information that comes partly from security (user profile) and partly from business data.

Our Model in SCLP

Here is the representation of the brokerage account access in SCLP.

```
/*
SCLP Rulebase in IBM CommonRules SCLPfile format: acctauth
Version: 1
Date: April 29, 2004
Author: Chitro Neogy, MOT 2004, Sloan School of Management <chitro@sloan.mit.edu>
*/

/* the following group of rules are created by the brokerage company,
for "RETAIL" and "CORRESPONDENT" clients */

/**
Default rule is not to grant access **/
/*
<defaultrule>
if setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company, ?RRonAccount, ?RR2onAccount)
and
    setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company) and
    CNEG accessGranted(self, ?Account, ?Requester);
*/

/** general rule - if account and employee types are different, access cannot be granted
Thus if accounttype is "RETAIL" and employee is "CORRESPONDENT", access is denied **/

<typerule>
```

```

        if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_acct, ?RRonAccount, ?RR2onAccount) and
    setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_emp) and
    notEquals(?acctType, ?empType)
    then
        CNEG accessGranted(self, ?Account, ?Requester);

```

```

/* "RETAIL" accounts - owned directly by brokerage company */

```

```

<retailAcct>

```

```

        if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_acct, ?RRonAccount, ?RR2onAccount) and
    setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_rep) and
    equals(?empType, "RETAIL") and equals(?acctType, "RETAIL")
    then
        accessGranted(self, ?Account, ?Requester);

```

```

/* "CORRESPONDENT" accounts - owned by "CORRESPONDENT" companies using brokerage
firm's computer system*/

```

```

<correspondentAcct>

```

```

        if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_acct, ?RRonAccount, ?RR2onAccount) and
    setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_emp) and
    equals(?acctType, "CORRESPONDENT") and equals(?empType, "CORRESPONDENT")
    and BranchRulesMet(?Requester, ?Account) and RegRepRulesMet(?Requester, ?Account)
    then
        accessGranted(self, ?Account, ?Requester);

```

```

/* Branch of account matches branch of requester. */

```

```

<branchAcctRule1>

```

```

        if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_acct, ?RRonAccount, ?RR2onAccount) and
    setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_emp) and
    equals(?company_acct, ?company_emp) and
    equals(?acctType, "CORRESPONDENT") and equals(?branchOfEmp, ?branchOfAcct)
    then
        BranchRulesMet(?Requester, ?Account);

```

```

/* Branch of requester has supervisory powers over bank of account. */

```

```

/*

```

```

<branchAcctRule2>

```

```

        if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_act, ?RRonAccount, ?RR2onAccount) and
    setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_emp) and
    setReqSubBranchList(?subBranchListOfRequester) and
    equals(?acctType, "CORRESPONDENT") and
    isInList(?branchOfAcct, ?subBranchListOfRequester)
    then
        BranchRulesMet();

```

```

*/

```

```

/**
If reg rep flag is set, then check id of user matches RR or RR2 field in account
*/
<regrepRule1>
  if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_acct, ?RRonAccount, ?RR2onAccount) and
  setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_emp) and
  setRepFlag(?company_rr, ?regRepFlag) and
  equals(?company_acct, ?company_emp) and
  equals(?company_acct, ?company_rr) and
  equals(?acctType, "CORRESPONDENT") and equals(?regRepFlag, 1) and
  (equals(?RRonAccount, ?Requester) or equals(?RR2onAccount, ?Requester))
  then
    RegRepRulesMet(?Requester, ?Account);

/**
If reg rep flag is not set, reg rep rules are automatically met
*/
<regrepRule2>
  if
setAcctParms(?Account, ?acctType, ?branchOfAcct, ?company_acct, ?RRonAccount, ?RR2onAccount) and
  setEmpParms(?Requester, ?empType, ?branchOfEmp, ?company_emp) and
  setRepFlag(?company_rr, ?regRepFlag) and
  equals(?acctType, "CORRESPONDENT") and equals(?company_acct, ?company_emp) and
  equals(?company_acct, ?company_rr) and equals(?regRepFlag, 0)
  then
    RegRepRulesMet(?Requester, ?Account);

/**
override definitions
*/
overrides(typerule, correspondentAcct);
overrides(typerule, retailAcct);
overrides(typerule, branchAcctRule1);
overrides(typerule, regrepRule1);
overrides(typerule, regrepRule2);
overrides(retailAcct, defaultrule);
overrides(correspondentAcct, defaultrule);

/**
Test Data - Accounts first
*/
setAcctParms(XYZ301098, "RETAIL", XYZ, FMR,0,0);
setAcctParms(301301099, "CORRESPONDENT", 301, BankAmerica, Roger, 0);
setAcctParms(201301098, "CORRESPONDENT", 201, Barnett, Trevor, 0);

/**
Company Rep Flag info
*/
setRepFlag(BankAmerica, 1);

```

```

setRepFlag(Barnett, 0);

/**
Employee Rep Info
*/
setEmpParms(Chitro, "RETAIL", XYZ, FMR);
setEmpParms(Roger, "CORRESPONDENT", 301, BankAmerica);
setEmpParms(Justin, "CORRESPONDENT", 301, BankAmerica);
setEmpParms(Trevor, "CORRESPONDENT", 201, Barnett);

```

```

SCLPEngine: Adorned Derived Conclusions:
accessGranted_c_2(self, XYZ301098, Chitro);
CNEG accessGranted_c_1(self, XYZ301098, Roger);
CNEG accessGranted_c_1(self, XYZ301098, Justin);
CNEG accessGranted_c_1(self, XYZ301098, Trevor);
CNEG accessGranted_c_1(self, 301301099, Chitro);
CNEG accessGranted_c_1(self, 201301098, Chitro);
BranchRulesMet(Roger, 301301099);
BranchRulesMet(Justin, 301301099);
BranchRulesMet(Trevor, 201301098);
RegRepRulesMet(Roger, 301301099);
RegRepRulesMet(Trevor, 201301098);
accessGranted_c_3(self, 301301099, Roger);
accessGranted_c_3(self, 201301098, Trevor);
accessGranted_u(self, XYZ301098, Chitro);
accessGranted_u(self, 301301099, Roger);
accessGranted_u(self, 201301098, Trevor);
CNEG accessGranted_u(self, XYZ301098, Roger);
CNEG accessGranted_u(self, XYZ301098, Justin);
CNEG accessGranted_u(self, XYZ301098, Trevor);
CNEG accessGranted_u(self, 301301099, Chitro);
CNEG accessGranted_u(self, 201301098, Chitro);
accessGranted(self, XYZ301098, Chitro);
accessGranted(self, 301301099, Roger);
accessGranted(self, 201301098, Trevor);
CNEG accessGranted(self, XYZ301098, Roger);
CNEG accessGranted(self, XYZ301098, Justin);
CNEG accessGranted(self, XYZ301098, Trevor);
CNEG accessGranted(self, 301301099, Chitro);
CNEG accessGranted(self, 201301098, Chitro);

```

SCLPEngine: Processing ends.....

5.2.4 Conclusion from SCLP examples

The general takeaway from these complex examples is that RuleML and SCLP can be used very effectively to represent system access rules in financial systems instead of having such policies embedded in proprietary applications. In reality,

we could use sensors to obtain (i.e. via query) critical data like account and employee parameters (in Case III) from different databases within the enterprise. At this point, vendors are attempting to implement the Check 21 system using a variety of proprietary techniques. Our recommendation is to reduce future complexity, vendor lock-in and interoperability issues by standardizing trust management rule representation through a common, scalable policy language. Our solution here is built in IBM Commonrules, which is based on Situated Courteous Logic Programs [16]. In future, we may be able to use RuleML, which offers a greater degree of interoperability and flexibility than any commercial rule language.

6. Conclusion

6.1 Research Findings

As we have seen in this thesis, business rules are pervasive across the financial services industry. Accurate representation and execution of such policies are critical to building trust in IT applications and business processes. Yet, in many corporations, 99% of business rules are embedded deep inside stovepiped applications, making them difficult to understand and expensive to maintain. The justification for continuing to develop software this way is it is quicker to initiate new projects, reuses existing people skills within the organization and eliminates the need to learn complex policy languages from rule vendors.

For companies worried about Trust Management, silo applications are particularly dangerous because of two reasons. First, access rules are embedded in applications and there is little confidence that policies are being evaluated or executed accurately. This is a significant exposure for the customer, especially given the steep penalties of regulatory compliance violations. Secondly, proprietary rule solutions lead to vendor lock-in problems, which can prove very costly for the customer. Such experiences also affect trust policy software vendors, since the customer is reluctant in making follow-up purchases, and the market remains small.

A number of small companies offer competing products for the Business Rule Engine (BRE) market but these are all based on proprietary technologies. Despite business benefits they bring, customers are hesitant to adopt such point solutions for mission critical applications because of vendor lock-in possibilities, especially with smaller vendors. The BRE market is relatively mature and the leading vendor products are not highly undifferentiated within the four families of CCI rule engines. Under these circumstances, we posit that developing rule engines on an open, standardized platform will provide benefits for the customer community and the solution providers overall by growing the market. The argument derives from the hypothesis that rule technology adoption will accelerate once clients are assured about greater interoperability and low switching costs, thereby generating greater revenues for lead vendors across the industry. While conventional wisdom in software business is to protect market share with proprietary solutions, the nature of bottlenecks in the BPM/BRE industry leads me to believe that the benefits of core functionality standardization will far outweigh the risks of commoditization (See Chapter 4 for details).

My research identifies a number of business drivers within the financial sector which create new opportunities for enterprise application development using standardized rule methodologies. One such technology is RuleML, whose powerful expressive features, prioritized conflict handling capabilities, tractability and clean procedural attachments make it an ideal candidate as a standard in the

rule industry. Since RuleML tools are relatively immature, we have used an earlier implementation of similar concepts called IBM CommonRules. We have provided several examples of business problems from the financial services industry, like Check 21 and Credit Card Authorization, and illustrated that the underlying complex rules can be represented elegantly using Courteous Logic Programs (CLP) in IBM CommonRules. The same application would have required more complexity to develop in a traditional proprietary language. CLP RuleML programs are also more human-readable, thereby improving senior management visibility into process controls, allowing faster policy upgrades and developing greater confidence in accuracy and audit of policies in financial systems.

One of the frequently cited causes of not using standardized rule engines is performance concern. It is true that generally, rule engines are a little slower than optimized, compiled code. However, the Total Cost of Ownership (TCO) of flexible rule systems is so much less that we can easily make up response time issues by compensating with additional, high-performance hardware. Another objection to standards is based on self-interests of commercial institutions that often back a certain standard. The origins of RuleML within a largely commercially neutral academic community including MIT and its adoption by a wide group of both non-commercial and industrial researchers are likely to allay concerns around standards dominance by specific players only.

As companies begin to realize greater ROI from RuleML, they will be motivated to buy more rule-based products and reengineer legacy applications around open standards. The BRE and BPM vendor community will respond by building RuleML compatibility within their product base. A RuleML based software can then be interoperable not only with other, compliant vendor rule engines but can also be integrated within institutional applications using standardized interfaces. This reinforcing positive feedback loop will unlock the huge potential of the BRE market and fuel rapid rule industry growth. While we have not performed a detail study on the authorization software market in this research, we do know from interviews that the industry is quite saturated with proprietary solutions. It would be interesting to see whether standardization can help to enhance demand of access control software products.

This thesis makes a contribution to the academic and scientific communities in several ways. To the best of my knowledge, it is the first study of contemporary trust-policy related applications and their characteristics in the overall financial services industry that can benefit from rule technologies. A second contribution lies in justifying the positioning of RuleML as the relevant primary, open standards policy language, both for rule-engine vendors and financial institutions. A third contribution is developing a set of Courteous Logic Program application scenarios to demonstrate the effectiveness of RuleML in business applications like Check 21 and credit card authorizations. The final contribution of this research is

proposing a rule based Trust Management model in security by arguing that RuleML can serve as a robust reference implementation of eXtensible Access Control Markup Language (XACML), an OASIS standard for digital authorization.

6.2 Suggested Future Work

One of the key motivators for BRE vendors to adopt RuleML will be business demand for standardization. My research analyzes and argues in favor of the need for open standards in financial industry rule systems. We encourage similar studies for other business sectors, especially in healthcare, pharmaceuticals, electronics and automotive industries. Even within the financial industry, the demanding timeline of this research only permitted study of a few verticals like banking, brokerage, credit cards, mutual funds and institutional portfolio companies. Study of other areas like capital markets and insurance can lead to even greater insights.

More mature reference implementations of RuleML are in the process of development, yet there are a number of practical rule engines that share commonalities with SCLP systems. Developing a RuleML compliant software package and optimizing it for performance benchmarking would be a great contribution to those interested in developing applications in RuleML for the financial industry. A related study would be to research and discover additional

requirements that will enhance the flexibility of RuleML in solving a wider array of problems.

This research makes the argument that standardization will lead to greater benefits for both consumers and rule engine service providers, especially for trust management in the financial services industry. It would also be desirable to test this hypothesis further through additional research. One example would be to perform a deeper analysis for just a certain business within financial services to describe the impact of using RuleML for developing business applications and buying RuleML compliant vendor solutions for meeting specific needs. In principle, by estimating or measuring the fractional cost of development, maintenance, system consolidation after mergers, compliance benefits, etc from greater use of rule systems across a firm's value chain, it would be possible to estimate quantitatively the ROI and payback periods that organizations can expect by adopting RuleML technologies. In a related manner, it will be good to have a greater analysis of trust policy software markets, especially for access control.

Finally, as the title of this thesis implies, another important direction is to link this technology back to the vision of the Semantic Web. Rule Standardization is critical to knowledge representation in the web world. RuleML can play that role, but we need multiple research activities to express rules in an agent-driven web architecture. We need to inject simplicity into our model, so that policy writing

becomes as straightforward as generating HTML code. With RDF and Ontologies already in place, the creation of a standard rule language will complete the major building blocks of the architecture and encourage the lead user communities to accelerate the ultimate buildup of the Semantic Web vision.

7. Acknowledgements

Writing this thesis, as well as joining Sloan after twelve years in the industry, required courage, perseverance and patience. Some of the strength came from within me, but perhaps more came from my own family. I am very fortunate to have a supporting family – my wife Ruma, my father Dr. Rajat Kumar Neogy and my two lovely boys, Chirantan (8) and Rupayan (7) – that recognized the priceless value of MIT education and were willing to make many personal sacrifices during the course of this busy year. I am truly grateful to God for the most wonderful gift in this world, the love of a caring family.

I would like to thank my MIT thesis supervisor, Professor Benjamin Grosf, for taking on this challenging project with me, and providing exceptional guidance at every turn. He unraveled the wondrous world of rule engines during our many discourses, and introduced me to SCLP that became the backbone of this thesis.

A number of individuals, recognized below, have contributed valuable input and shared key insights for our work. I would like to thank all of them for their generosity in sharing time, resources and information for this research.

- ◆ Andy Elliott, Pega Systems
- ◆ Arup Ray, Numeric Investments
- ◆ Deepak Verma, eCredit.com

- ◆ Hal Lockhart, BEA Systems and OASIS XACML TC
- ◆ Henry Ancona, Pega Systems
- ◆ Horace Henderson, Numeric Investments
- ◆ Jay Butler, Fidelity Investments
- ◆ Joseph Wilkinson, Intellisphere
- ◆ Lalana Kagal, PhD student, University of Maryland
- ◆ Paul McNulty, Pega Systems
- ◆ Prateek Mishra, Netegrity
- ◆ Ray Chang, Fidelity Investments
- ◆ Ray Graber, Ray Graber and Associates
- ◆ Said Tabet, McGregor and the RuleML Initiative
- ◆ Shirley Kawamoto, Independent Security Consultant
- ◆ Susan Bates, Fleet Bank
- ◆ Vivek Pandit, Independent Management Consultant

Finally, I want to thank two very special friends, Mr. Partha Ghosh, Senior Advisor at Monitor, Boston and Mr. Raphael Carty, Vice President at JPM Chase, NY. Both of them encouraged me to apply to MIT and were kind enough to provide outstanding references on my behalf. Having learned from their leadership examples, someday I hope to inspire other deserving students to achieve greater things in life and offer them the best possible assistance to realize their dreams.

8. References and Bibliography

1. Bird Graham B., "The Business Benefits to Standards", StandardView, Vol 6, No. 2, June 1998
2. Blaze M., J. Figenbaum, J. Lacy, "Decentralized Trust Management", Proceedings of IEEE Conference – Privacy and Security, 1996
3. Check 21 Resource Document,
<http://www.nacha.org/Check21%20Resource%20Document.pdf>
4. Christensen Clayton M., "The Innovators Dilemma", Harper Business, 2002, pp 44 - 47
5. Coleman, J. S., Foundations of Social Theory, Cambridge, MA, Belknap Press, 1990.
6. DAML Services, www.daml.org/services
7. Davenport, Harvard Business Review, Business Value of IT, Chapter – "Saving IT's Soul", Harvard Business School Press, 1999, pp 10-11
8. Decker Stefan, Sergey Melnik, Frank Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, Ian Horrocks, "The Semantic Web – the roles of XML and RDF", IEEE Internet Computing, Sep 2000, pp 63-74
9. Denker Grit, Lalana Kagal, Tim Finin, Massimo Paolucci and Katia Sycara, "Security for DAML Web Services – Annotation and Matchmaking", Second International Semantic Web Conference (ISWC), Sep 2003.

10. Farkas Csilla, Michael N. Huhns, "Making agents secure on semantic web", IEEE Internet Computing, Nov 2002, Vol 2, pp 76-79
11. Finin Tom, Anupam Joshi, "Agents, Trust and Information Access on the Semantic Web", SIGMOD Record, Vol 31, No 4, Dec 2002
12. Foray Dominique, "Standardization and Innovation in technological dynamics", StandardView Vol 6, No 2, June 1998
13. Forrester Tech. Spending Summary Brief: Financial Services, June, 2003.
14. Fukuyama, F, "Trust: the social virtues and the creation of prosperity", The Free Press, New York, NY, 1995.
15. Gordon L., M. Loeb, "The Economics of Information Security Investment", ACM Transactions on Information and System Security, Vol 5, No 4, Nov 2002, pp 438-457
16. Grosf Benjamin , Hoi Chan, "Introduction to CLP Rules Programming", IBM Web Site and Commonrules Guide, Aug. 1999.
17. Grosf Benjamin, "Representing E-commerce rules via Situated Courteous Logic Programs in RuleML", Accepted Journal Paper for Electronic Commerce Research and Applications, revised version of Sept. 2003.
18. Grosf Benjamin, "SweetDeal – Representing Agent Contracts with Exceptions using XML Rules, Ontologies and Process Descriptions", Presentations at the 12th International Conference on WWW, Budapest, Hungary, May 2003

19. Grosf Benjamin, Harold Boley, "Introduction to RuleML- Teleconference meeting of joint US/EU markup committee meeting", Oct. 2002.
20. Grosf Benjamin, Mahesh Gandhe, Tim Finin, "SweetJess: Inferencing in Situated Courteous RuleML via translation to and from JessRules", Working Paper, May 2003.
21. Grosf Benjamin, Yannis Labrou, Hoi Chan, "A declarative approach to business rules in contracts – courteous logic programs in XML", http://ebusiness.mit.edu/bgrosf/paps/ec99_proceedings.pdf
22. Health Insurance Portability and Accountability Act of 1996, <http://www.hhs.gov/ocr/hipaa/>
23. Hwang, P. and W. Burgers, "Properties of Trust: An Analytical View", Organizational Behavior and Human Decision Processes, 69(1), 67-73, 1997
24. Information Security magazine, Cover Story, AAA of information Security – Authentication, Authorization, Accounting, Oct. 2000, <http://infosecuritymag.techtarget.com/articles/october00/coverb.shtml>
25. Kabbaj Mohammed Youssef, "Strategic and Policy Prospects for Semantic Web Services Adoption in the US online travel industry", MS Thesis in Technology and Policy, MIT, June 2003
26. Kraft Reiner, "Designing a Distributed Access Control processor for Network Services on the Web", Proceedings of the ACM Workshop on XML Security, 2003, pp 36 – 52

27. Lee Tim Berners, "Semantic Web Introduction", talk on W3C day, Tokyo, Japan, 2003, <http://www.w3.org/2003/Talks/1113-sw-tbl/slide1-1.html>
28. Lee Tim Berners, Hendler, Lassilla, "The Semantic Web", Scientific American, 2001
29. Leebaert, Derek (editor), "The Future of the Electronic Marketplaces", MIT Press, 2nd edition, 1999, pp 145 - 173
30. Li Ninghui , Grosz, Figenbaum, "Delegation Logic – a logic-based approach to distributed authorization", ACM Transactions on Information and System Security, Vol 6, Issue 1, Feb 2003
31. Lockhart Hal, Presentation on "Authorization Infrastructure – a standards view", OASIS & BEA Systems, 2004.
32. Luhmann, N., "Trust and Power", John Wiley and Sons, London, 1979
33. Mayer, R. C., Davis, J. H. and Schoorman, F. D., "An Integrative Model of Organizational Trust", The Academy of Management Review, 20, 709-34, 1995.
34. McIlraith Sheila, Tran Cao Son, Honglei Zeng, "Semantic Web Services", IEEE Intelligent Systems, March 2001, pp 46-53
35. Meissner L. P., "Who pays for standards? Who should play for standards", ACM SIGPLAN FORTRAN Forum, Vol 14, Issue 1, March 1995

36. Neogy Chitravanu, Akio Saita, Renato Catalan, Paulo Vita, Boon Chung, Yasushi Iguchi, 15.379 MIT class Presentation from Financial Services Technology Group, Fall, 2003.
37. NetworkWorldFusion, Top Web Services Worry – Security, Jan 2002, http://www.nwfusion.com/news/2002/0121webservices.html?doc_id=7747
38. Newcomer E. , “Understanding Web Services”, Addison Wesley, 2002
39. O’Neill Mark, et al., “Web Services Security”, McGrawhill/Osborne, 2003.
40. Oasis Working Committee on XACML, “eXtensible Access Control Markup Language”, Working Draft 05, January 2004, location - <http://www.oasis-open.org/committees/xacml/repository/oasis-xacml-2.0-core-spec-wd-05.pdf>
41. Park Joon, Ravi Sandhu and Gail-Joon Ahn, ACM transactions on Information and System Security, Vol 4, No 1, Feb 2001, pp 37 – 71
42. Preece, S. Decker, “Intelligent Web Services”, IEEE Intelligent Systems, Feb 2002.
43. RuleML Web Site, www.ruleml.org
44. Sarbanes Oxley IT costs, ComputerWorld Article, June 2003, <http://www.computerworld.com/managementtopics/management/itspending/story/0,10801,87446p2,00.html>
45. Sarbanes Oxley, http://www.aicpa.org/info/sarbanes_oxley_summary.htm
46. SEC web site, “Circuit Breakers and other Market Volatility Procedures”, April 2003, <http://www.sec.gov/answers/circuit.htm>

47. Shurmer Mark, Gary Lea, "Telecommunications Standardization and IP Rights – a fundamental dilemma?", StandardView, Vol 3, No 2, June 1995.
48. Sinur J. , J. Thompson, The Gartner Group, "BPM Pure Play 2003 Magic Quadrant", Research Note, June 2003
49. Sinur J., The Gartner Group, "Drivers for BPM", Research Note, Feb 2004.
50. Sinur J., The Gartner Group, "The Business Rule Engine 2003 Magic Quadrant", Research Note, April 2003
51. Sultan F., G. Urban, V. Shankar, I. Bart, "Determinants and Role of trust in E-business – a large scale empirical study", MIT Working paper 4282-02, Dec 2002.
52. Symantec Internet Security Threat Report, Volume V, March 2004.
53. The Gramm-Leach-Bliley act, <http://banking.senate.gov/conf/>
54. The New Capital BASEL Accord, <http://www.bis.org/publ/bcbsca.htm>
55. The US Patriot Act, <http://www.epic.org/privacy/terrorism/hr3162.html>
56. Utterback James M., "Mastering the Dynamics of Innovation", Harvard Business School Press, 1994, pp 90 - 101
57. World Wide Web Consortium (W3C), www.w3c.org