December 2003

Model for Trust Among Peers in Electronic Multiparty Transactions

Sanjay Goel
University at Albany, SUNY

Jagdish Gangolly
University at Albany, SUNY

MODEL FOR TRUST AMONG PEERS IN ELECTRONIC MULTIPARTY TRANSACTIONS

Sanjay Goel
University at Albany/SUNY
goel@albany.edu

Jagdish S. Gangolly
University at Albany/SUNY
gangolly@albany.edu

Abstract

We address the issue of trust among the principals in a multiparty transaction in a peer-to-peer system implementing a service-based architecture. We propose a model that allows peers to form instant federations and collectively consummate complex business transactions. A transaction is done in two phases. In the first phase services are selected and a federation is formed. In the second phase the services are executed based on the precedence relations defined in the transaction. The peers are 'passionate' entities during the first phase. For this phase a transitive trust model is used wherein trust among distant peers is established with each other using a chain of intermediaries via mutual acquaintances. Once the federation is formed, the entities become rational and a secure environment is created to execute the transaction. For this phase a distributed trust model using simple public key infrastructure is proposed.

Keywords: Trust, peer-to-peer systems, security, graphs, metrics

Introduction

There are several peer-to-peer networks that are in existence today and have gained popularity with users wanting to share their files. The most notable example is Napster (2003) that had 70 million subscribers sharing data and bandwidth on a P2P basis. Napster used a central file server that contains the index of all the files that each node is contributing at the time it joins the network. Even though Napster is not a true P2P system (it uses a fixed server to connect peers together), it has demonstrated the feasibility of a large-scale implementation of a P2P-type system. Freenet (Clarke 2000) and Gnutella (Oram 2000, Gnutella 2003) have gone a step further in allowing people to share computational resources as well as network capacity. In Gnutella, each node acts as a router making routing decisions, thus using some of its CPU. It has 10 million subscribers. Kazaa (Good and Krikelberg, 2003) is the current leader in P2P systems and has become the hotbed for sharing of all kinds of documents including music, video, books etc.

† *Assistant Professor of Management Science & Information Systems*
‡ *Associate Professor of Accounting and of Management Science & Information Systems*

P2P is not really about sharing music, nor is it a collection of tools and applications. It is just a computing concept. It defines all decentralized distributed components in the system in that any machine on the network can take on the role of either a client or a server, depending on the need for a service or its capability to provide a service. These peer components might be devices, computers or objects on the network.

Most of the P2P systems discussed so far have successfully allowed exchange of information. Peer-to-peer networking is becoming popular in distributed computation as well. The **seti@home** project, however, allows people to perform computation in the address space of different computers. Jini (Kumaran 2002) is another distributed architecture that allows creation of spontaneous network of P2P services. Jini defines the protocol for discovery and join and introduces a concept of leasing to make the network self-managing and resilient. The main problem with the P2P systems described above is that they lack interoperability. JXTA (Gong, 2002) makes P2P interoperable by defining the basic communications protocols in XML. It can operate over a variety of network protocols (e.g. Bluetooth, TCP/IP) and route across Network Address Translation (NAT), Dynamic Host Configuration Protocol (DHCP) and firewalls. JXTA works by mapping the physical networks of systems, Network Address

Translations, firewalls, and protocols into a virtual network where every peer is connected to every other peer. A uniform addressing scheme makes this possible, such that every peer has a unique ID. There are numerous other XML based initiatives to allow ubiquitous exchange of data via XML and support web service that have been summarized at the W3C web site (w3c 2003).

In the work on multi-party transactions, peers are distributed network objects of the same type. Each object has the same top-level interface, however, the implementation of the interface may be different for each object. Rather than the currently prevalent client/server model, in which all communication passes through and is controlled by a central server (Web, FTP, mail, application servers and the like), in P2P systems the communication goes directly from one user's or virtual enterprise's object to another user's or virtual enterprise's object. Since accessing these decentralized object nodes, called the grid, means operating in an environment of unstable connectivity and unpredictable IP addresses, the grid nodes must operate outside the Domain Name System and also have significant or total autonomy from central servers.

As additional computing nodes enter the network, the network capacity grows and the resources of the new members are added to the network, thereby increasing its computing power and resulting in a self-scaling process. A peer that requires services broadcasts its requests to the other peers. The peer that can provide the service offers to process the request. There is still a concept of a service provider and a service requestor that is akin to the server and the client. So what makes a peer a peer? The trademark of a client/server system is that a server has a fixed IP-address for the host on which it runs and this enables the clients to be able to find it. The second mark of a client server system is that the server and the client have fixed specified roles with respect to each other.

On the other hand, the identity of the service provider or the service user is not central to a peer-to-peer system. A P2P system is characterized by nameless, faceless entities that work together using standard interfaces for completing complex tasks. As long as they adhere to a given set of rules (common interface), they can collaborate with peers and participate in transactions. What makes a peer a peer is who any peer can be seamlessly replaced by another peer that implements equivalent business logic without re-implementing or modifying the client. A service provider may become a service requestor for any service and, similarly, a service requestor may become a service provider for a different service. This lack of identity that characterizes these services makes the issue of trust critical to the success of a peer-to-peer system.

Edfandiari and Chadrasekharan (2001) have defined three mechanisms for establishing trust, observation, interaction and institutions. The observation model works by capturing the uncertainties in different agents and then using a Bayesian learning model to compute the trust. The interaction model is based on use of standard protocols between agents akin to the network protocols. The institutional model is based on the aggregation of the performance history by a trusted third party. Josang (1996) has more formally categorized the agents as passionate and rational based on their behavior. If the decision-making of the entity (like humans) is based on free will, it is called passionate entity. Entities like algorithms, protocols, software are called rational since their behavior is controlled by a set of rules rather than free will. Golbeck et. al. (2003) have defined a metric for computing trust in social networks.

The paper addresses the issue of trust and security in P2P systems among the principals while executing multiparty transactions. Section 2 describes the alternate models for trust that can be used and describes the rationale for selection of the models. It describes trust models for both passionate and rational entities. Section 3 describes the simulation that is underway for observing the evolution of trust in a P2P system. Section 4 concludes the paper and discusses the future directions of our work.

Security and Trust

Security and trust are essential requirements in design of electronic transactions for B2B/B2C systems. The multiparty transaction described in this work is completed in two phases. In the first phase, the services are selected to form a federation and, in the second phase, the services in the federation perform operations to complete the transaction. In the service selection phase, the peers are passionate and base their trust on the reputation of the peers. In the second phase, the entities become rational and create a secure environment to protect the data as well as the integrity of peer systems. Distributed systems have certain unique issues as opposed to the traditional client/server systems. The subsequent paragraphs present a discussion of these issues and provide a feasible solution for establishing trust in the context of a service based architecture in a P2P environment. Section 2.1 discusses the issue of trust between peers during the process of service selection and presents a model of trust for passionate entities. Section 2.2 discusses the issue of trust in a federation of peers and presents a model for securing the transaction as well as protecting the integrity of peers.

Trust in Service Selection

Trust is the expectation in a transaction by each party that the other party will deliver on its promises. Trust is usually ascertained by the past performance history of an organization, or through validation by a third party that is trusted. In brick and mortar companies, trust is established by the brand name recognition that a company develops. In the currently operating web-based businesses like *Amazon.com, Nytimes.com* and *Yahoo.com* which use the traditional client/server technology, the identity of the parties involved in a transaction is already known and trust is established on the basis of name recognition. In the eBay model, the trust should not only exist on the reputation of the provider but also among the peers who are involved in the transactions. The first level of trust comes from the name recognition and the trust among the peers is based on the validation from other peers. In P2P systems supporting a service based architecture, multiple peers are allowed to implement the same interface. This allows a seamless integration of services, whereby one service can be transparently substituted with another service.

For a P2P system within an intranet the peers be may automatically trusted to deliver on their promises because of organizational controls. On the public internet however, establishing trust among peers that lack identity either requires a trusted third party to validate the quality and performance of the peer or the transaction history of the service provider from which its performance can be deduced.

Centralized Trust

To implement trust via a trusted third party, the transaction history of the service provider would need to be tracked and by using an evaluation metric, the performance of the provider would be scored. A simple rating metric, based on the feedback from existing transactions can be a ratio of successful and unsuccessful transactions. If the entity participated in a federation that successfully completed a transaction the rating would go up and vice versa. The main problem with this approach is that a distributed system is relying on a centralized service creating a point of failure for the system and a potential bottleneck in the system.

The third party rating scheme can be implemented using PKI whereby the rating agency would issue digital rating certificates. In this scheme, the rating agency computes the trust rating based on past performance and issues a rating certificate to the service provider for each service it offers. The service provider presents the rating certificate to the service requestor in order to establish trust with the requestor. The rating certificates are akin to the X.509 digital certificates that are used for verifying the public key of the certificate holder. Also, just as in the public key infrastructure (PKI), a chain of trust can be built wherein the rating authority delegates to local rating authorities. This approach suffers from all the fundamental problems that PKI suffers from, i.e., complexity, cost, and large maintenance overhead. In a dynamic system where services are entering and leaving the network and the rating is changing dynamically, keeping the rating scores up-to-date would be expensive and maintaining revocation lists infeasible. A distributed model of trust for is thus proposed as an alternate to the schemes presented above.

Distributed Trust

The distributed trust model is based on the Small Worlds phenomenon in social networks which suggests that each person is linked to every other person in the World by a short chain of intermediaries. Milgrim's (1967) studies have shown that each person in the United States is connected to every other person by a median of 5.5 intermediaries. We use the same phenomenon for establishing trust among the peers. If two peers do not know each other there must be a chain of intermediary peers that trust each other through which these peers must be related. Thus, if each neighbor can establish trust with a small set of neighbors they are able to establish trust with other peers on the network, based on a transitive relationship. The trust evolves over time as the services use more services in the network and get connected through a chain of services. If we assume that A trusts B and B trusts C then the transitive model states that A trusts C. The question is how much does A trust C and how does such trust evolve over time.

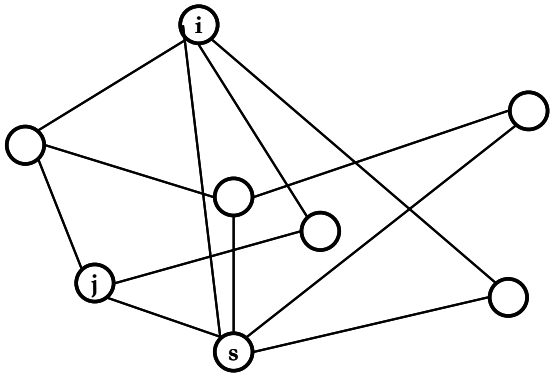

Figure 1. Schematic Showing the Nodes of a Network

This model can be studied using graphs where each peer can be represented as a node on a graph, where trust relationships are represented by edges between nodes. When trust between two entities not directly connected to each other is to be computed, a chain of intermediaries is identified and trust is computed based on the level of trust between intermediaries in the chain. The question that needs to be answered is: how to compute trust if there is more than one path between the two vertices and whether the graph is connected. In other words, is it always possible to reach one vertex from another vertex via a chain of intermediaries?

To address the issue of trust between two nodes connected by multiple paths a general case is considered where two nodes are connected by a series of intermediaries. For example consider the graph shown in Figure 1. The nodes in that graph correspond to peers in the network and each edge in the graph represents the trust relationship between the two peers. Let the edge between nodes i and j have a value τ_{ij} ($0 < \tau < 1$) which is the trust of node j by node i. The trust probabilities between two nodes i and s can be computed by using a simple equation shown in equation 1 below (Golbeck, 2003).

$$t_{is} = \frac{\sum_{j=0}^{j=n} \begin{cases} (t_{js} * t_{ij}) & if \quad t_{ij} \geq t_{js} \\ (t_{ij}{}^2) & if \quad t_{ij} < t_{js} \end{cases}}{\sum_{j=0}^{j=n} t_{ij}} \qquad \text{Equation 1}$$

Where, i has n neighbors with paths to s, and t_{ij} is the trust between nodes i and j. In this equation only two-edge paths are considered, and trust is computed by taking an average of all the two-level trust links between nodes i and s.

It is intuitively clear that the graph would not always be connected, for instance when a new service node first enters the network and has no history of transactions it will not be connected to any node. Such nodes will either try to enter with a reference to some other nodes (like belonging to the same subnet representing an organization, or being provisioned by existing services) or will start with a low rating before they establish their performance history.

The bucket brigade algorithm first proposed by John Holland (1962) for reinforcement learning in classifier systems is used to model the evolution of trust in the network. If a peer is a part of a federation that successfully consummates a transaction, the trust of that peer as well as all the peers that were intermediaries in the trust chain rises. Similarly if a peer is a part of a federation that failed to consummate a transaction, trust of all the peers that were intermediaries in the trust chain lowers. All the peers in the federation develop a direct connection with each other leading to an increase in the connectivity of the network. Also, when a peer leaves the network, all the edges that are connected to it are also destroyed. Trust is activity dependent; that is, a large period of inactivity either in becoming a part of a trust chain or actively participating in a federation leads to a decrease in trust. Over time the trust in inactive providers keeps getting lower until they atrophy and die.

At the inception of such a network the services have links to their immediate neighbors and the graph of the network is close to a regular graph. However, as new service federations are formed by discovery of new services, links are established between the nodes of the federation. As new links are formed randomly between the peers, average path length between pairs of nodes reduces. According to Barabasi et. al (Barabasi, 1999) networks observed in peer-to-peer systems exhibit power-law distribution of edges per node. This means that the number of edges per node can be approximated by the power law equation (*edges = c x nodes*). In this network, the active nodes with high trust and high connectivity should keep getting more connected while the weaker nodes that fail to participate in federations or become intermediaries in the trust chain, gradually atrophy, and eventually, disappear from the network. An agent based simulation model is being used to investigate the evolution of trust in such a network. This will be described in section 3.

Once the federation is formed, a secure environment is required for executing the transaction. The next section describes the model used for creating a secure transaction environment.

Rational Trust

Once a federation of peers is formed all the peers behave as rational entities. The transaction now requires, a secure computing environment that spans across multiple address spaces. The four main issues related to security design are authentication, authorization, confidentiality and integrity. P2P systems present some unique security challenges compared to the traditional

client/server model in addressing these security needs. The first issue is a lack of identity (name recognition) in the peers. The second issue is the execution of mobile code, that is, migration of code from one peers object space to another peers object space where it executes. Thus to ensure a secure transaction, identity of each party in the transaction needs to be verified, secure communication needs to be established, access to resources needs to be authorized, and activities of the mobile code need to be constrained within the parameters of the transaction.

P2P systems are characterized by standardized interfaces that can be implemented by service providers. This allows multiple service providers to implement the same service and allows the clients to substitute one service for another service seamlessly. Service providers can join any transaction, disperse after its completion and then join a different transaction. Authorizing the providers to access confidential data in such an environment when the shared resources do not have fixed associations requires a dynamic security system whereby authorization decisions are automatically made based on the attributes of the principals involved.

Whenever mobile code (which travels from one address space to another) is used, there is a need to prevent the code from performing illegal operations in the host address space where it is executing. Mobile code brings up some unique issues (Davis Project 2003), viz., mutual authorization, proxy trust, and data integrity. Mutual authorization means it is not only the service provider who decides what it will allow the client to do but also what the service user will allow the downloaded proxy to do on its machine. Proxy trust implies the user trusts that the proxy actually belongs to the service it says it does. Data integrity means that the data is not modified accidentally or maliciously while in transit over the network.

These issues need to be addressed at the design level in a P2P system. The subsequent subsections address the issues mentioned above and describe how the issue of confidentiality, authentication and authorization are handled in a distributed system.

Confidentiality and Authentication

Confidentiality (Privacy) and Authentication are orthogonal. Confidentiality prevents unauthorized disclosure of contents, whereas, authentication verifies the identity of the requestor and the integrity of the message contents. Confidentiality is usually achieved by moving data that is cryptographically protected. Despite the complexities and the overhead associated with digital certificates, they are the only viable choice for establishing identity of the owner. Digital certificates are issued by Certificate Authorities (CA) and contain identifying information of the owner, the public key of the owner and the identity of the certificate authority. The CA signs the certificate using its private key, thereby, vouching for the identity of the public key owner. A certificate authority can delegate its authority to another party by certifying the public key of that party, which in turn can similarly delegate its authority to another party by signing its public key, thereby building a chain of certificates. While verifying a certificate, the tree of certificates is traversed to make sure that the certificate is authentic. When two parties interact, one or both of the parties need to verify their identity using digital certificates. When a user and a provider interact, the provider sends its digital certificate to the user. When two providers interact they exchange their digital certificates with each other. For secure communication a session key is exchanged and a secure session is established. To facilitate the exchange of a session key, service providers need to possess a digital certificate which allows them to transmit data using asymmetric key encryption.

Authorization

In a secure system, access to shared resources needs to be protected from unauthorized access. The principal (a user or a provider) must be explicitly authorized to access specific resources based on the level of trust. Security is administered by verifying the authorization of the principal prior to access of any resource available to the providers. Access Control Lists (ACLs) contain a list of principals that are allowed to access a specific resource. Such lists are maintained for each resource that needs to be protected. Demurjian, et al. (2000) have developed a role-based security architecture using ACLs in the Jini based distributed environment for a university enrollment application. Even though they are cumbersome, ACLs provide a feasible solution to provide security in a static environment where the associations between shared resources are clearly defined and a centralized trusted authority exists. In very dynamic distributed environments ACLs have a very large maintenance overhead associated with them. The rapidly changing associations between the parties need to be reflected in the ACLs which though feasible, can be a cumbersome task.

Simple Distributed Security Infrastructure (SDSI) developed by Ron Rivest and Butler Lampson at MIT (1996) is another suitable mechanism to provide security in a distributed environment. Simple Public Key Infrastructure (SPKI) developed by Ellison, et al. (1996, 1999) is also a flexible authorization model. These two designs were merged together and called the SPKI/SDSI (Ellison

2001) model. This model provides an elegant solution for security in a distributed environment. In this environment each peer is a certificate authority and can issue a certificate. Thus, a matrix of certificates is created without a hierarchy and without a trusted root certificate authority. There are two kinds of certificates: name certificates and authorization certificates.

The name certificate binds a name to a public key. A name certificate defines a local name in the certificate issuer's local namespace and does not have to ensure that the name is unique in a global namespace making it scalable. Local namespaces can be linked together by binding a name in one object's namespace to another provider's namespace. A group is created when a peer issues name certificates to other peers by signing the certificates using its own private key. This allows for an easy maintenance of mutual authorizations since the entire group can be given an authorization for a given resource. In addition, since the certificates issued have an expiry time, temporary groups can be created whenever multiple parties come together for transacting on the web. In order to maintain integrity of the PKI, digital certificates may be revoked in case of suspicion that a private key has been compromised. Certification Revocation Lists (CRLs) are lists of all such revoked certificates. These lists must be checked prior to acceptance of any digital certificate. There are no CRLs in SPKI, however, the expiry time is set so that the certificate expiry time coordinates closely with the completion of the transaction. There is an issue that the time of expiry may not match exactly with the completion of the transaction; however, the period for which the authorization remains after the transaction should be fairly small and this should not impact the long-term security of the system. A provider can be in multiple groups depending on the providers that interact with it and the resources that it needs access to.

Authorization certificates are issued to provide specific parties access to resources which have restricted access. For instance, in a business-to-business transaction, the vendor may authorize the buyer access to the inspection database on a batch of parts being manufactured. Delegation of authorization is also possible using these certificates. During a transaction, if one of the providers in the network needs support from another provider it can delegate authorization to that provider. The provider who received the authorization by delegation can further delegate authorization to another provider, and hence a chain of trust can be built. The authority to further delegate the authorization is controlled by the certificate issuer who can set the delegation bit of the certificate to true or false. Such certificates based on SPKI have been used by Eronen and Nikander (2001) in the Jini-based secure network infrastructure developed for mobile devices.

Simulation

The peer-to-peer network is being simulated using an agent-based simulation model (Repast 2003). In this simulation the peer-to-peer network is treated as a two-dimensional graph with the peers as the nodes of the graphs. The nodes are connected with arcs if there is trust between any two nodes; and they are not connected if there is no trust between the nodes. The value of the arc signifies the level of trust between any two nodes. Based on the trust between the nodes, requests for multiparty transactions are generated. To process the requests, process of *discovery* is used to create a federation of services. The transactions are processed and the trust between nodes evolves as the simulation progresses. The transactions succeed and fail using a uniform distribution. Also new service providers and requestors enter and leave the network periodically. Using the metric of trust defined earlier in the paper, the evolution of trust is computed. In addition, the attributes of the graph, that is, clustering and path length, are monitored over time.

Conclusions

The authors investigated the issue of trust among peers in multiparty transactions using distributed systems. These transactions go through two phases. In the first phase, the parties are passionate and peers are selected to form a federation based on the trust among the entities. In the second phase, the entities become rational when they are trying to create a secure environment for the transaction. The authors have presented a scheme where trust is transitively obtained by forming chains of trust through mutual acquaintances. At the second level the authors present a scheme for ensuring trust among rational entities using Simple Public Key Infrastructure. The authors plan to pursue this work further by using graph models coupled with agent based simulation and investigate different scenarios of trust evolution.

References

Barabási, A., Albert, R., and Jeong, H., Mean-field Theory for Scale-Free Random Networks, Physica A 272, 173-187 (1999) (available online at **http://www.nd.edu/~networks/papers.htm#paper3**).

Clarke, I., Sandberg, O., Wiley, B., and Hong, T. "Freenet: A Distributed Anonymous Information Storage and Retrieval System," In Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, 2000.

Davis Project. See **http://davis.jini.org**. Visited on March 21, 2003.

Demurjian, S., "Role-Based Security in a Distributed Resource Environment," Proc. of Fourteenth IFIP WG 11.3, Working Conference on Database Security, Scoorl, The Netherlands, 2000.

Ellison, C., "Establishing Identity Without Certification Authorities," 6th USENIX Security Symposium, 1996.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T., "Simple Public Key Certificate," The Internet Society, 1999.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T. "RFC 2693: SPKI Certificate Theory," The Internet Society, 1999.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T. "RFC 2693: SPKI Requirements," The Internet Society, 1999.

Ellison, C. SPKI/SDSI and the Web of Trust. See **http://world.std.com/~cme/html/web.html**. Visited on March 21, 2003.

Eronen, P., and Nikander, P. "Decentralized Jini security," Proceedings of the Network and Distributed System Security Symposium, San Diego, California, 2001.

Esfandiari B. and Chandrasekharan S., "On How Agents Make Friends: Mechanisms for Trust Acquisition", Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies 2001, pp 27-34.

Golbeck J., Hendler J., Parsia B., Trust Networks on the Semantic Web, WWW 2003, May 20-26, 2003, Budapest, Hungary.

Gong L., JXTA: A Network Programming Environment, Internet Computing Online, May/June, Vol.5, No. 3, pp. 88-95.

Good, N.S., and Krekelberg, A. "Usability and privacy: a study of Kazaa P2P file-sharing." See **http://www.hpl.hp.com/shl/papers/kazaa/KazaaUsability.pdf**. Visited on March 21, 2003.

Gnutella website. See **http://gnutella.wego.com**. Visited on March 21, 2003.

Holland, J. H. "Outline for a logical theory of adaptive systems," Journal of the Association Computing Machinery (Vol. 3), 1962, pp. 297-314.

Java Doc. See **http://java.sun.com/products/jdk/1.2/docs/guide/security/Policy/Files.html**. Visited on February 14, 2003.

Josang, A. Proceedings of the 1996 workshop on New security paradigms, Lake Arrowhead, CA, 1996.

Kumaran S. I., Jini Technology An Overview, Prentice Hall Ptr, 2002.

Napster Home Page. See **http://opennap.sourceforge.net**. Visited on March 21, 2003.

Oram, A. Gnutella and Freenet Represent True Technological Innovation, The O'Reilly Network. **http://www.oreillynet.com/lpt/a/208**. Visited on March 21, 2003.

Rivest, R., and Lampson, B. "SDSI - A Simple Distributed Security Infrastructure." See http://theory.lcs.mit.edu/~rivest/sdsi10.ps. Visited on March 21, 2003.

Sirbu, M., and Chuang, J. "Distributed Authentication in Kerberos Using Public Key Cryptography," Proceedings of the Internet Society Symposium on Network and Distributed System Security NDSS'97, San Diego, CA, 1997.

W3C website **http://www.w3.org/2000/03/29-XML-protocol-matrix**, visited March 20, 2003.