Integrating Reputation Services into emerging Web Services Architecture

Ramesh Babu
Jim McGovern

School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
e-Email: r.babu@computer.org, jim.mcgovern@rmit.edu.au

Abstract

Successful business interactions in e-commerce need integration between service providers and service requestors based on mutual trust. Reputation is a form of trust in a social context. Formal reputation models currently do not fully meet the semantic integration needs in emerging Web Service Architectures. An integrating model for a Reputation Service is proposed to mediate in such electronic interactions. The model decouples the service requestor from provider and does not mandate any changes or strict rules of interaction. The model also offers solution to the semantic integration problem and is suitable to evolve with standards.

Keywords

reputation, trust, web services, security, e-commerce, decoupling, integration, mediation

INTRODUCTION

Trust and reputation are important considerations in e-commerce. Trust for clients means that service providers will use their information responsibly and will fulfil their service obligations. For service providers it means that those who receive services are genuine, conform to rules and also meet their obligations in using the service. In the absence of experience and a history of trust between parties, the reputation of each party becomes important factors for participation and conduct of e-commerce transactions.

Berners-Lee et al. (2001), W3C (2002a, b) position Web Services Architecture as transforming the Web from a collection of documents into a distributed computational platform that can provide a wide range of information processing services. Ankolekar et al. (2002) refer to initiatives such as ebXML and the Semantic Web provide information descriptions, ontology and security policy languages that complement this development. Web Services Architecture is still in its infancy and there is considerable scope in completing the requirements, especially in relation to reputation and trust.

Conceptual models of reputation and isolated reputation systems have been developed and some implemented (see for example, Maxmilien and Singh (2002); Mui et al. (2002), Lethin (2001)), however, there is also much scope for improvement, particularly in the light of Web services development. Many reputation systems to date are based on a centralised model and do not exploit the potential for distribution in the Internet. Some are embedded in particular services such as eBay. Others are limited, being restricted to particular metrics, scoring algorithms or domains.

This paper describes a new integrated model for reputation services, and the main aspect of how it will work as a Reputation Service in the Web services context. The remainder of the paper is organised as follows. First, a broad background on the developments in Web services is presented. Second, existing reputation models are briefly discussed. Third, the idea of a Reputation Service in a Web services context is introduced and the considerations that influence the development of a conceptual model are discussed. Fourth, the *"integrating model"* for a *reputation service* is described with a comparison to related developments that show the "integrating" strength of this model. Finally, functions, elements and properties of the reputation service architecture are presented for future work.

WEB SERVICE ARCHITECTURE

W3C (2002b) define Web services as software systems identified by a URI (Uniform Resource Identifier), which can be matched against XML descriptions of services. Users interact with these services using XML based messages conveyed over the Internet. The vision of the Web Services Architecture is to create a self describing system of Web-based services. W3C (2002b) primarily has three types of entities; service requestors, discovery agents and service providers. Requestors find services from published information provided by

requestors to the discovery service. Requestors get enough information to invoke (either dynamically or statically) the services using suitable message exchange protocols. Discovery agents rely on registries containing information about services to mediate in the Web Service Architecture. The primary infrastructure of Web Services Architecture consists of a registry, service description language and service access protocol.

The registry defined by Universal Data Description Interface or UDDI holds information about businesses, their services, service categories and technical information required to invoke these services. The Web Service Description Language (WSDL) is an XML based interface description that defines objects types and elements exchanged and input output definitions. The XML based object and service access protocol called Simple Object Access Protocol (SOAP) is a method for message passing and function invocation across standard protocols such as HTTP and SMTP. Service descriptions are in XML and fall into two categories, service interface descriptions and service implementations. Service interface descriptions are reusable descriptions for a class of services. Service implementations refer to service interface definitions and point to their realisations in an end-point URL.

Decoupling is a primary principle of Web Services Architecture in Fensel and Bussler (2002). Mediators or Discovery Agents decouple requestors and providers by allowing heterogeneity and restricting to interface dependency between requestor and provider. Services can be implemented, configured, changed independent of service usage as long as interface contracts are maintained. URI makes everything in locatable reference context. Requestors do not need to make custom changes to use the interface apart from agreeing to use the interface. Providers may keep the implementation proprietary and confidential to gain competitive advantage.

The follow on benefit is *scalability* from this mediation. Complex software integration is not necessary and mediators also make the entire infrastructure scaleable allowing several requestors to converse with several service providers.

Web services already support extensions and features such as guaranteed delivery, signatures, and encryption, which are essential to trust and reputation, but it also provides the framework for implementing a wide range of new services, including reputation services.

EbXML

EbXML is an emerging global standard for business exchange in a setting where an overall business process and business roles and protocols for such exchange are agreed to by the parties (http://www.ebxml.org). Patil and Newcomer (2003) highlight the commonalities between Web services and ebXML. Web services use a bottom-up approach based on technologies such as SOAP and XML as well as the mapping to wire protocols to enable transport, package and extensions to handle transactions and security. EbXML uses a top down approach to define an overarching business process, business workflow, agreements and protocol for document exchange with a view to controlling and managing the business process between the parties. The ebXML model needs a registry of parties, services and actions, an underlying messaging service, a mechanism to advertise the capabilities of collaborating parties, their agreements, and definition of the business documents used in exchanges. Due to the top-down approach, ebXML includes higher level semantic concepts and capture these in a well-defined architecture. EbXML also separates wire protocol from description of entities and types in the model and a top layer for discovery and semantics. EbXML is a standard very much in development like Web Services Architecture, and greater integration of the two approaches seems likely over time.

Semantic Web

The Semantic Web refers to self describing web sites and services with content that has attached semantics and context or properties that can be understood in machine language suitable for automation. Currently the web is a large collection of inter-related documents, which a human needs to interpret and understand. It is not easy for this to be done automatically by software programs or agents. McIlraith et al. (2001)argue that pervasive, ubiquitous, autonomous web service agents need semantics to be attached to the web to support intelligent interaction.

Technologies and standards are being developed to make the Semantic Web initiative, a reality. The Resource Description Framework Schema (RDF-S) supports self-describing documents. RDF-S offers a set of richer representation modelling primitives based on classes (or definition of a concept), sub-classes, properties and domain/range restriction on properties. To describe e-commerce interactions complex representations and formalisms are needed. For example, models of entities, activities, agreements, shared meanings, and obligations are needed as concepts within a conceptual framework of shared meanings.

Fensel et al. (2001) define Ontology as *formal, explicit specification of a shared conceptualization*, is a promising means of addressing these descriptions. Horrocks and Patel-Schneider (2003) list several issues that still remain in unifying the different representation of concepts, knowledge and logic formalisms.

Ontology extends the registry-based approach defined in Web Services Architecture and ebXML. Ontology Inference Layer (OIL) is a sematic web initiative that will add capabilities on top of RDF-S to include necessary mainstream modelling primitives such as classes and attributes found in frame-based systems. Frame based systems are generally better at modelling real-world semantics compared to simple predicate based systems. OIL extends RDF-S by storing and comparing resource definitions and properties as well as handling more complex scenarios with additional representational and reasoning abilities found in expressive rule based and logic based languages such as DAML-S. These extensions provide the ability to exchange information with autonomous agents, encode and enforce policy thereby giving it the ability to automate security functions.

EXISTING REPUTATION MODELS

In a distributed computing environment as exemplified by Web Services, a customer/client's choice of a service provider should not be based only on the advertised functionality of the provider's service. The service may not be available, responsive or have the necessary quality at the time the service is needed. Although these non-functional "attributes" can be advertised as well, the client may not be aware of the past history of the provider in executing service promises. Mechanisms such as a "reputation service" would be able to track past histories of use, recommendations, community feedback and so on to predict or assert future behaviour or quality with certain degrees of confidence.

Reputation implies that there is a community or web of trust and interaction between members of the community. The reputation of a member in a community is a collective score or trust value assigned to the trustor and/or trustee. The reputation is also dynamic and it could become the basis for decision making for co-operation in a community (or reciprocity in a bi-lateral relation).

This section examines notable features of several existing reputation models in developing our model. The models address different aspects with some common themes. Aspects include developing and maintaining a history of interactions, feedback or scoring based systems, recommendations, use of reputation related attributes, use of metrics and scoring, working within the context of a domain, quality of service metrics as attributed to Web Services, and temporal aspects of reputation.

Reputation History

Lethin (2001) defines reputation as the abstracted memory of past transactions, while Abdul-Rahman and Hailes (2000) similarly define reputation as "*an expectation about an agent's behaviour based on the information about or observations of its past behaviour"*. A reputation service *tracks delivery against promise/offer* of provider's services, and constructs this history of trust. Clients, either manually or automatically, should be able indicate their satisfaction with a service and, conversely, providers should be able to respond to any adverse commentary of their service provisioning.

Time is an important dimension in establishing reputation. Reputations can be built or destroyed. Providers should be able to recover from "bad" reputations or unusually aberrant behaviour. In taking account of history, older events should be down-valued or "damped" with recent stable history given more importance in determining reputation. Again, this cannot be done without context to the meaning of the interaction. Resnick et al. (2000) show how reputation systems such as eBay help build trust and balance the buyer and seller requirements in a market using feedback to guide trust decisions. Without such reputation measures only low quality sellers will remain in a market. However, the eBay reputation system is not an open architecture service.

Mui et al., (2002) show a mutually reinforcing relation between reputation, trust and reciprocity and use a probability measure for reciprocation. Reputation is defined as "*perception that an agent creates through past actions about its intentions and norms"*. The model shows that trust can be computed given the history of encounters and reputation as an expectation function. This model approximates "complete strangers" with a uniform distribution which may not match all contexts and puts forward a propagation mechanism for reputation in a society of interactions. This model suffers from a clear definition of context and does not include the environmental factors in interactions.

Recommendations

Abdul-Rahman and Hailes, (2000, 1997) use recommendations and trust graphs and Lethin (2001) uses endorsements as important aspects in establishing reputation. Agent behaviour is accumulated in a history of

encounters and a recommender trust value is assigned. Assessing the performance of recommenders is based on individual experiences.

These models have some limitations. Trust and recommendation usage context is passive and does not capture trust intentions, objectives, trust mechanisms an environment valuation. Trust accumulation over several encounters is taken as a simple maximisation function where the "best experience" dominates. There is a need to clearly model "experiences" for example as hierarchies. Good and bad experience may have different properties. Experience may relate to objectives and the impact may vary. There are also difficulties in clearly understanding the intentions of recommenders unless the objectives or properties that recommender values most are known. A "bad" recommendation from a recommender does not mean the recommender is harmful; there may have been a difference in 'perspectives' between recommender and a potential user of the recommendation.

Reputation measures

The reputation of a member in a community is a collective score or trust value assigned to the trustee. Maxmilien and Singh (2002) define reputation as the aggregation of reputation ratings of an individual. Ratings are defined as the vector of attribute values, where attributes are domain specific variables relevant to reputation. This model is more suitable for agent architectures where each agent defines the suitable attribute aggregation and where the context of use is limited. There exists a *reputation index* (or scoring mechanism), which ranks the reputation of providers against a client's intended usage context and preferences. Preferences are important, as assessment of reputation is subjective, and one user of a service may use different set of attributes to determine reputation from another. The reputation index should incorporate the damping and smoothing of historical data to provide a fairer assessment of reputation. The reputation is also dynamic and it could become the basis for decision making for co-operation in a community (or reciprocity in a bi-lateral relation).

Ran (2003) proposes an extension to the UDDI registry to implement a QoS (Quality of Service) metric model, which requestors can query to match preferred criteria. The implementation of semantics within a registry could be challenging due to the metadata nature of the registries. Registries in general do not hold context specific or domain specific information. Also, the registry interfaces should be modified to cater for the needs of the more complex QoS semantics.

ESSENTIAL ELEMENTS OF REPUTATION MODELS

Existing models provide essential elements of a reputation model needed for the problem of trust, however, there are deficiencies. In some cases, different computation models are used (such as probabilistic, logic or rule based, subjective estimates). Models used are assessed based on their suitability to narrow application domains and no unifying models or characteristics emerge. A comprehensive model will use many of the essential elements of successful models. An approach that extends these models using some underlying integrating principles that support evolving standards can provide a better basis for developing reputation services.

The service should be widely accessible and robust. It should be willing to participate with unknown entities gradually building trust, differentiating between clients and providers. Reputation service, as a community, need to have high availability and be able to withstand attacks.

There exists a "contract" between a client and a provider, which is the basis for judging successful or unsuccessful delivery of service. The contracts bring additional semantics to the interactions, modelling the rights and obligations of the parties. Whenever a client requests information from a reputation service, it provides an intended usage context, which is likely to be part of the client's side of any client-provider contract. These contracts are also the basis of processes for dealing with complaints. Clients have no basis for complaint if they use a service differently from that which is advertised. Providers are subject to complaint if the use of their service does not deliver results as advertised. Clients making a complaint need to provide "strong" evidence supporting their complaint. Providers may be rated better if they are 'open' to criticism and help streamline grievance procedures.

The reputation service operates under several limiting uncertainties and complexities. First, there is ***user uncertainty*** regarding identity, motives, credentials for each interaction and semantic context of use. Strategies of co-operation may fail when interacting with others who are selfish, where as it may be in self-interest to be able to share information with a wider community to face such situations. Parties interacting on an economic basis may find it ***computationally complex*** to determine the cost/benefits of interactions and or define the exact nature, amount or opportunity cost of risk in interactions. Another issue is the sheer number of interactions across multiple, ubiquitous interfaces. This is ***integration complexity***. Interactions may occur across stateless or state maintaining interactions. Another issue is the inherent limitations of parties. It is almost impossible for any party to completely maintain exact state and history of all past interactions. This is ***history complexity***. Finally, the multitudes of interfaces involving humans, businesses, non-businesses, governments and other roles,

automated software agents interacting for different purposes across multiple inter-related or unrelated domains give us the issue of ***semantic complexity***.

It is difficult to maintain a single view of trust and reputation across all these interactions as well as apply this information to direct or manage the interactions. Two interactions between the same parties may vary greatly on the semantic context depending on what is being exchanged. For example, the failure to logon due to an incorrect password may not be treated as severe as a rejection of a payment transaction signed and approved by some party. It is important to differentiate the different usage scenarios and "understand" how the different scenarios impact the trust / reputation model.

The position of "trust" (and therefore, reputation) in interactions needs to rise to the semantic layer. A framework is needed where several different meanings can co-exist and interoperate without reducing the overall meaning or usefulness of free interactions. To facilitate this discovery a set of principles are proposed as principles for the "integrating" model:

- ***Scalable mediation*** - This principle is derived from the web services architecture framework. This implies that the reputation service must be able to scale exponentially to handle several mediations including policy management, computations, and collaboration with directories, performing critical analysis aware of constraints, rights, obligations and preferences of interacting parties.
- ***Decoupling*** - This is another principle derived from web services architecture framework. Decoupling means the architecture 'facilitates' meaningful interfaces not requiring that components be glued to each other directly. The business intelligence of the implementing services and interactions for usage of the service are separated using an interface layer using information hiding principles. The motivations for this are two fold. Firstly, it allows least changes to interacting requestors, when service implementations and underlying intelligence mechanism changes. However, the decoupling also hides the complexity and intelligence that allows one to implement trust policies without requestor's knowledge and keeps the motivations of the reputation service secure from prying.
- ***Adaptive Behaviour*** – This principle helps supporting of multiple interfaces, personalities (trust) and detecting and accommodating preferences. This implies that one cannot expect situational normalcy always, and should not depend on the environment to be fail-safe. Instead, one tunes and adapts to the environment and reacts intelligently to these changes.
- ***Measure, extend and demand Trust*** – This principle shows the openness in the interactions but with a measure of purpose. In measuring trust, one needs to understand the obligations of parties and measure that the fulfilment meets or exceeds these expectations. Extending trust shows that a newcomer is allowed to participate or progress through a trust building phase. Demanding trust is to identify the core concepts of trust such as integrity and ability and expect these in interactions and use the progress on these dimensions to control risky interactions.
- ***Fairness*** – one can only expect trusting behaviour and attitude when it is reciprocated or made possible in the environment that should support it. For example, if the reputation service is not fair to providers if two equally merit the consideration and one is favoured over the other. This could lead to dissatisfaction and eventually less trust and participation. Usually any managed environment is policy driven and it is important to consider fairness in policy. Fairness is not only desirable, but should be computable. This is an advanced notion and needs further research. For example, it is not clear whether there is any relation between fairness and trust.
- ***Proactive*** – It is better to understand and be aware of the imperatives and actively pursue actions to better one's position. Being proactive is better than being reactive. However, policy considerations may override this default principle.
- ***Efficiency*** – Services should strive to become more efficient in relation to systems and conversations – which mean economic use of network connections, processing cycles, memory, and storage space.
- ***Active Delegation*** – In carrying out the services it is inevitable that in distributed computing, one needs to delegate trust across different channels to parties, delegate the facilitation of tasks in the mediation process, delegate resolution of issues, delegation of provision of information, delegation of carrying out actions. However, effective delegation cannot be carried out without regard to trust considerations. It seems important that there is a trust chain pre-requisite to forming a delegation chain.
- ***History or Memory*** – Effective use of history is an important aspect of providing a reputation service. History is a core requirement of providing context. However, the model should be flexible to allow bootstrapping interactions where no history exists. It is expected that such interactions will be constrained, but the protocols should allow trust to develop.

- ***Measurable Actions, Outcomes*** – Related to the semantics of the interactions are measurable outcomes and measurable actions (obligations) of parties. Each interaction that is worth occurring needs to be measured and will ultimately decide the Trust and Reputation.
- ***Context evolution, top-down, bottom-up*** – In interactions the semantic context may not be clear with the available information. However, with correlation between interactions and references, the higher or lower level context can be derived. This principle states the dynamic nature of the context in its use of reputation service model.

REPUTATION SERVICE - THE INTEGRATING MODEL

Conceptual models for reputation service will co-exist with similar models for electronic business (e-business) and e-commerce. Several formal models for electronic business are available. Jones et al. (2000) define participants, roles and their activities. Shankar et al. (2002) look at how elements of a trust model can be defined by examining value propositions of stakeholders in the business. The essential model proposed here encompasses and extends the several business models found elsewhere and the more recent developments in the Semantic Web Services and EbXML context.

The essential model can be described as an "integrating" model. Each model in the composite serves an orthogonal need (a dimension). The model implementations may be allocated to orthogonal subject areas with defined interfaces for integration into the reputation service. Models interact with each other in the context of semantic web service architectures, with protocols and security as required for such interactions.

The model needs to serve several aspects for functioning within the Web Services Architecture context. "Figure 1 - Main aspects of a Reputation Service" shows the aspects of *Interface*, *Transform*, *Remember* and *Interpret*. *Interface* and *Transform* are support layers interacting with external services and entities. *Remember* and *Interpret* are semantically richer layers. The granularity of the implementation will depend on the scope of the Reputation Service, which can range from a small agent to a large scale service portal that serves multiple domains, clients and agents.

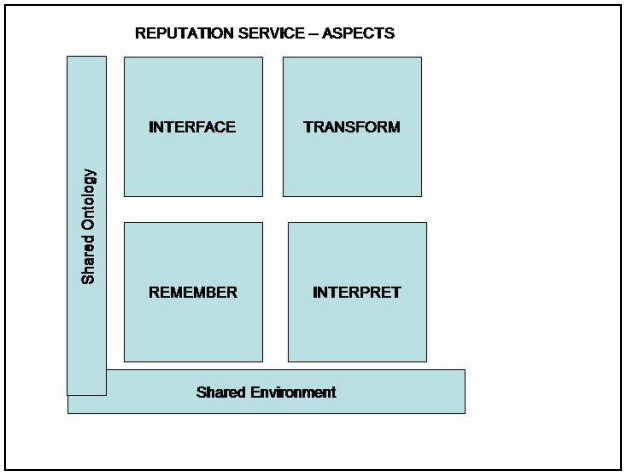

Figure 1 - Main aspects of a Reputation Service

All aspects of the service should operate in a shared environment with a shared ontology. The term "shared" does not imply any shared control of what is being shared, but limited to sharing by means of interactions and a proximity due to channels of communication that exist. *Decoupling* is a core principle in this model as described before. Other e-business models have tight integration between model components. *Interface* will abstract and only "remember" pointers to additional context implemented elsewhere. Therefore, the entire model is not monolithic. The models are distributed, decoupled, federated in nature. Ontology is shared and discovered not dictated and hardwired in the model. Sharing in general is done in a resource framework, where every resource is identified, be it a model resource or a real world business resource for consumption. It has properties, location reference, security properties and history. Such decoupling allows for best use in each modelled subject area and allows evolution of the reputation service as semantic web implementations progress.

The *Remember* aspect contains a persistent model described in "Figure 2 - Reputation Service – The Integrating Model". The model consists of *Entities* that have some *Utilities* that can be assessed using an *Analysis* model. The *Entities* themselves offer services defined by *Service* models that can be realized with formal or informal contracts. The interactions or *Activities* engaged in by the *Entities* then relate in the context of these *Contracts*.

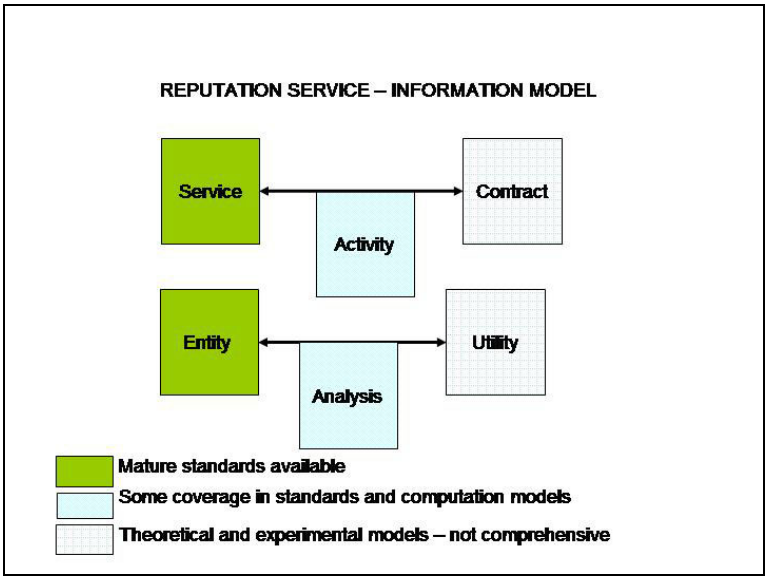

Figure 2 - Reputation Service – The Integrating Model

A further detailed description will relate each of the constituent models to other existing developments so far in the paper.

Entity Model

The Entity model defines the different parties in e-commerce interactions. All stakeholders and/or their proxies need to be defined here. E-business stakeholders include customers, businesses (service partners), shareholders, distributors, suppliers, employees, governments, regulators, financial institutions and so on. This model captures the relationship, characteristics and any implied hierarchies. It is also possible to attach ontology and reputations to these entities.

Utility Model

The Utility model holds the objectives, purposes, business goals, policies, preferences and choices of stakeholders in the e-commerce system. This holds economic efficiency requirements at various levels (organizational or individual transactional, environmental etc). Individual biases and characteristics regarding utilities and trends may also be available here for analysis. The model needs to hold different types of resources (money, time, and other costs).

Analysis Model

Various computational models of trust and reputation can be accommodated, for purpose of utility. The Analysis model embodies reasoning based on the history of encounters. It understands types of trust elements, obligations, assurance characteristics, and risk attitudes. This model implementation is also decoupled, and can accommodate specialised analysis services from different sources. The Analysis model needs to keep track of the core trust dimensions, systemic trust parameters, trust dispositions, deceptive trends, feedbacks or recommendations, maintain history of encounters and consider systemic efficiency. This model is central to the reputation service and needs to interact with various other models such as Activity, Service, and Entity as well as Utility models.

Service Model

The Service model is defined along the lines of the web services model. A service has characteristics, types exchanged, interface properties, quality of service parameters, pre-requisites, pre-conditions, post-conditions. Services may be composed of other services (for composite services), orchestration (for extended business choreography) workflow defined by an associated Activity model. Parts of the Service model will be able to be implemented in a mature UDDI Registry and some related parts in an ebXML Registry.

Contract Model

The Contract model stands for modelling interaction intents, licenses, contracts, and collaboration protocol agreements (as in ebXML). The model captures properties, rules (pre-requisites or pre-conditions or service instance override conditions), protocols, non-disclosure agreements, use of information and third-party uses. Contracts come with the notion of measurement of service level agreement, service acceptance criteria, and financial incentives for good performance and penalties for poor performance. Contracts also imply rights and obligations. (Rights may exist independent of contracts, such as intellectual property, copyright, and license to use). Contracts will need to formally specify the value exchange between parties in the interaction and understand the impact to trust and reputation from the engagements.

Activity Model

The Activity Model is the realisation of service contracts and captures instances of interactions possible as defined in the service model. The Activity model captures policies, regulations, past actions and encounters. This will contain model workflow, list of responsibilities and roles in a workflow, work breakdown into tasks, event model, and initiation and termination rules.

RELATION TO WEB SERVICES ARCHITECTURE

The strength of this model is in the integration principle. It does not mandate changes to any of the existing standards, but puts forward a model for co-existence and evolution at the same time addressing the needs of reputation. For example, Entity and Service model implementations can be abstracted away into a UDDI registry model, whereas the Contract model may be partially implemented in an EbXML registry. Decoupling is also on an ontology and environment basis. It could be argued that the meaning of reputation scores may be meaningless outside a certain context – therefore, decoupling has its limits.

The table below shows a comparison between the model elements and implementations that exist currently. The comparison is approximate and will allow one to benefit from existing implementations.

	UDDI	EbXML
Entity	businessEntity publisherAssertion (a document centric relationship model between entities)	RegistryObject, Slot, Association, ExternalIdentifier ExternalLink RegistryPackage User, PostalAddress, EmailAddress, Organization
Service	businessService	Service, ServiceBinding
	bindingTemplate	
	tModel	
Activity	-	AuditableEvent, Messaging
Generic	tModel CategoryBags	ClassificationScheme, ClassificationNode, Classification
Contract	-	CPP, CPA
Utility	-	-
Analysis	-	-

Table 1 - Comparison of model elements

It can be seen that semantically rich components are modelled sparsely in mainstream Web Services Architectures; this will be a contribution of the proposed model. The proposed model will bring all required attributes complete with semantics and context in determining the reputation in interactions. The architecture will enable a learning semantics environment. The proposed model will achieve this co-existing with several parallel developments in standards and architectures with its ability to interface, transform and interpret.

REPUTATION SERVICE ARCHITECTURE

A reputation service is primarily a semantic Web service. Its principles, service properties, and functions may be outlined informally in the semantic web context. The technology used will support automatic agent processing and integration into the architecture.

The diagram below shows the context diagram of a reputation service in and some of the main functional components.

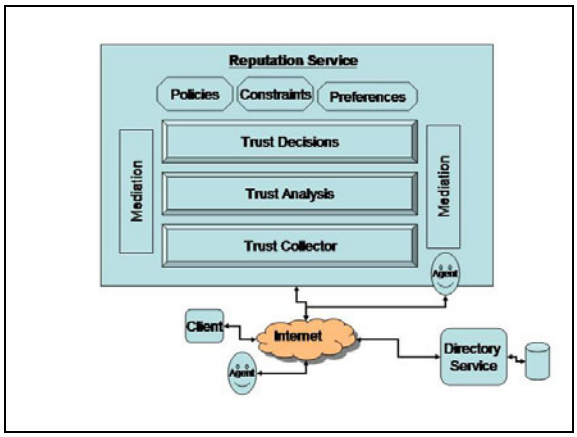

Figure 3 - Reputation Service Architecture

Primarily it has trust computation responsibilities. Trust computation needs, policies, constraints and preferences that drive adaptive behaviour and setting the environment context for the reputation service. Trust decisions are based on data collected while in active engagement with a requestor or another mediator. The Trust collector may also perform active data seek operations to collect history and recommendations as a listener. For example, client configurations may support agent hosting from favourite reputation services.

This section summarizes several functional elements or blocks that will exist in the architecture. For example, some examples are listed; a trust manager, an ontology mapping service, policy manager, role manager, and protocol handling. The mediation interface will need to host some of the elements such as protocol handling and ontology mapping. The mediation layer in addition to this will have a resource locator function, mediation locator function, workflow mediation involving service bridging and co-ordination of workflows supporting time based events. However, the mediation will be limited to reputation information.

Agent support is required to address intelligent agent adaptation. Agent factory function, roaming agent homing support and agent trust management will be required functions.

The Reputation Service can also expect peer groups which will need policy matching functions, reputation and information sharing. The Reputation Service will need to function as a web service with clients requesting and posting feedback, service quality information, service matching, directory updates, analysis aspects, algorithm matching, algorithm learning strategies, historical data, policy interrogations, volunteered statistics, independent analysis of service providers, service provider/service parameter matching, directory interfaces, value add directory features with trust, reputation, constraints, locale and bias information. In addition, the Reputation Service will need to serve human interfaces.

FUTURE WORK

More work is required to elaborate on the scenarios and protocols and expand on the model elements. The work on the architecture may be refined by showing explicit decoupling and mapping of models to external service elements with proper interface abstractions. To provide real value Reputation services need to become large scale deployments of mediated services, which addresses agent integration, proactive value add with mediation, discovery and negotiation.

Ultimately, the aim will be to implement and evaluate the model in a selected domain or domains with agents and human interactions to capture the different semantics. Several services will be defined in a simulated business workflow or other interaction requiring rich set of semantics, which need the use of logics to interpret reputation related functions.

REFERENCES

Ankolekar A, Burstein M, Hobbs JR, Lassila O, Martin DL, McIlraith SA, Narayanan S, Paolucci M, Payne T, Sycara K, Zeng H (2002) "DAML-S: Semantic Markup for Web Services". Electronic Commerce Research and Applications 2002

Berners-Lee T, Hendler J, Lassila O (2001) "The Semantic Web", Scientific American 284: 34–43

Fensel D, Bussler C (2002) "The Web Services Modeling Framework WSMF", Electronic Commerce Research and Applications, Vol 1: pp.113-137

Fensel D, Harmelen Fv, Horrocks I, McGuinness DL, Patel-Schneider PF (2001) "OIL: An Ontology Infrastructure for the Semantic Web", IEEE Intelligent Systems

Horrocks I, Patel-Schneider PF (2003) "Three Theses of Representation in Semantic Web", Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary 2003. ACM

Jones S, Wilikens M, Morris P, Masera M (2000) "Trust Requirements in E-Business" Communications of the ACM, vol 43, pp 81-87

Lethin R (2001) "Chapter 17: Reputation", In: Oram A (ed) Peer-to-Peer: harnessing the benefits of a disruptive technology, First edition. O'Reilly & Associates, Inc

Maxmilien EM, Singh MP (2002) "Conceptual Model of Web Service Reputation". ACM SIGMOD Record 31: 36-41

McIlraith SA, Son TC, Zeng H (2001) "Semantic Web Services" IEEE Intelligent Systems Vol 16: pp.46-53

Mui L, Mohtashemi M, Halberstadt A "A Computational Model of Trust and Reputation" Proceedings of the International Conference on Systems Sciences, Hawaii 2002

Patil S, Newcomer E (2003) "ebXML and Web Services" IEEE Internet Computing, vol 7, pp 74-82

Ran S (2003) "A Model for Web Services Discovery with QoS" ACM SigeCom Exchanges, vol 4.1

Resnick P, Zeckhauser R, Firedman E, Kuwabara K (2000) "Reputation Systems" Communications of the ACM, vol 43, pp 45-48

Shankar V, Sultan F, Urban GL (2002) "Online Trust and e-Business Strategy: Concepts, Implications, and Future Directions" eCommerce Research Forum, MIT

W3C (2002a) "Web Services Architecture Requirements", W3C Working Draft 14 November 2002.

W3C (2002b) "Web Services Architecture", W3C Working Draft 14 November 2002.

ACKNOWLEDGEMENTS

The authors thank Dr. Zahir Tari for his comments.

COPYRIGHT