Towards Accountable Enterprise Mashup Services

Joe Zou and Christopher J. Pavlovski
IBM Corporation, 601 Pacific Highway, St Leonard NSW Australia
joezou@au1.ibm.com, chris_pav@au1.ibm.com

Abstract

As a result of the proliferation of Web 2.0 style web sites, the practice of mashup services has become increasingly popular in the web development community. While mashup services bring flexibility and speed in delivering new valuable services to consumers, the issue of accountability associated with the mashup practice remains largely ignored by the industry. Furthermore, realizing the great benefits of mashup services, industry leaders are eagerly pushing these services into the enterprise arena. Although enterprise mashup services hold great promise in delivering a flexible SOA solution in a business context, the lack of accountability in current mashup solutions may render this ineffective in the enterprise environment. This paper defines accountability for mashup services, analyses the underlying issues in practice, and finally proposes a framework and ontology to model accountability. This model may then be used to develop effective accountability solutions for mashup environments.

1. Introduction

The recent and rapid expansion of Web 2.0 has placed considerably pressure upon industry to institutionalize new technologies and conform to emerging standards. While agreement on the scope of the term Web 2.0 does vary, O'Reilly provides a commonly accepted definition, noting this to include a range of enhanced services including web services, wikis, blogging, BitTorrents, and syndication [1].

The rapid growth of Web 2.0 has also introduced a number of new design patterns and architectural styles in web development. One of the notable techniques involves the mashing up information from existing services to deliver value-added new services. This process effectively involves the drawing of content from several sources to create a new content or service. The resulting web page is then referred to as a mashup of the existing content.

While mashup services bring flexibility and speed in delivering new valuable services to consumers, the legal implications of using this technology are significant. Researchers in law conclude that the development of mashup web services is fraught with potential legal liabilities that require careful consideration [4].

The issue of accountability associated with the mashup practice remains largely ignored by the industry. Current formal practices suggest that the mashup developer and original content source owner disclaim any warranties [4]. This appears to be temporarily acceptable since most services from Web 2.0 sites are free to internet users. This means that as long as consumers accept the terms and conditions, the issue of accountability is largely avoided. Notwithstanding, as these services mature to involve some payment, such an approach may no longer be tenable to all parties.

Traditionally, accountability implies that an entity has an obligation for the execution of authority and/or the fulfillment of responsibility [14]. Non-repudiation of transaction is also a major requirement for a service requester and service provider. However, in a mashup service scenario, the issue of accountability is more complicated. Firstly, there may be several implicit service providers involved due to the fact that the service is mashed up from a number of sources. Secondly, the content presented may not be delivered by the content originator. Furthermore, the sourced content may be altered or extended during the mashup process. Considering this problem further, does the body who modifies or augments the content assume entire liability for all the re-purposed content, including all accuracies and inaccuracies?

In this paper we consider that the accountability issue in mashup services is a broader and more complex theme when compared to non-repudiation in an eCommerce transaction. We propose a framework that includes the service or content creator as well as the new owner of the resulting mashed-up service. While accountability issues may not be fully addressable with the current technology, we believe

0-7695-3003-6/07 $25.00 © 2007 IEEE
DOI 10.1109/ICEBE.2007.12

that the first step towards enabling accountability in mashup services is to add more disclosure, trust, and un-deniability. This includes identities of all the parties involved and traceability in service composition. We also suggest that the concepts of involved parties and roles are essential in the service ontology model, such as in the Ontology Web Language for Web Services (OWL-S) model [3, 7].

Given that accountability in mashup has not been treated rigorously before, we view the main contributions of this paper as follows.

1. Analyze the underlying mashup accountability issues in practice;
2. provide a formal definition for accountability in mashup solutions; and
3. outline a framework and ontology to model accountability in mashup environments.

Using these tools it is hoped that more effective accountability solutions can be prepared on the basis of our framework and model. The remainder of this paper is structured as follows. The next section provides background information on the mashup paradigm and related work in the current literature in accountability. In section three we suggest a definition for accountability to address the additional requirements of mashup services. This is followed in section four by a framework and ontology that may be used to model accountability. Finally, we summarize our results and observations, discussing areas of further work.

2. Background and Related Work

2.1. Overview of Mashup Services

The term mashup originates from the practice of mixing song samples from two or more sources to produce a new sound track. In the context of the Internet, mashups are websites or applications that combine content from more than one source into an integrated application. This is generally achieved by using third party content provider application programming interfaces (APIs) or open technologies, for instance Ajax and PHP, and syndicated feeds such as RSS or ATOM. In addition, since the content may be obtained from several sources, intermediate businesses have emerged that act as content brokers. These intermediataries provide access to several content sources from 3rd parties, and also supply functions to support the mashup process.

Based on the concept of service composition in Service Oriented Architecture (SOA), mashup provides flexible and dynamic services with rich experience. This technology also enables a dynamic form of service reusability in contrast to the traditional method of static "cut & paste" reusability. However, since mashup involves the aggregation of another party's content into some new service or application, a number of legal issues are introduced [4]. When legal issues arise, accountability will become the critical concern for the parties involved.

The following diagram (Figure 1) illustrates the fundamental concepts in Web 2.0 mashups, where data and content is drawn from a range of sources to produce a new aggregated content or service. For example, the content may be drawn from local data repositories, from existing local and external web pages, accessed via SOA based APIs, and from intermediate content brokers (who source content from other parties).

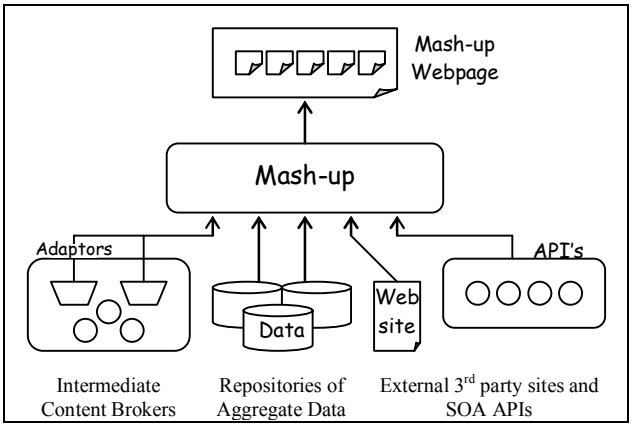

Figure 1. Mashup Services

While mashup applications and services are growing at a rapid rate, currently these appear to be applied in non mission critical services and are offered to the internet consumer largely as free services. In practice, the legal responsibilities are generally avoided by the content provider disclaiming all warranties and liabilities [4].

More recently, industry leaders are accepting mashup as an enterprise tool to enable the creation of so called situational applications. These types of applications solve business problems such as inventory management, sales and marketing information systems [5]. This emerging approach has been termed 'enterprise mashup' and several enterprise tools have been released [6]. Enterprise mashup may be viewed as a Web2.0 technology that builds upon the flexibility offered by SOA, and having a requirement for increased security.

Although enterprise mashup services hold great promise in delivering a flexible SOA solution in a business context, the lack of accountability in current mashup solutions may render this ineffective in the

business environment. As such, as more enterprises embrace this technology in building mashup services, the issue of accountability will manifest as a key concern for the service stakeholders.

2.2. Related Work on Accountability

The meaning of the term accountability appears to vary considerably and is dependant upon the context. Traditionally the topic of accountability has attracted much interest with focus on the eCommerce transaction. According to Kailar, accountability is "the property whereby the association of a unique originator with an object or action can be proved to a third party" [8]. The definition implies non-repudiation in an eCommerce transaction. Kailar also proposes a framework for the analysis of communication protocols that require accountability [8, 9].

Bhattacharya and Paul assert that while a digital signature can provide help in enabling accountability in two direct communication nodes, it can not fully address the accountability issues in multi-hop message due to the Sender's Ambiguity Problem [10].

In [11], the scope of accountability is broadened to represent the ownership of the responsibility to meet requirements in an end-to-end business process. The authors propose Accountability Centered Approach (ACA) for business process engineering. The ACA approach suggests iterative decomposition of accountability to appropriate levels and mapping of sub-accountabilities into activities.

A 3-D approach in accountability model (Detect, Diagnose, and Defuse) is proposed in [12] to discover and eliminate the root cause of problems when violations of service level agreement occur in business processes. The approach adopts Bayesian Network reasoning for root cause analysis and service reputation model to address problematic web services.

While existing research on accountability helps traditional eCommerce application and SOA business applications, the issue of accountability in service mashup has not been treated in the literature. In addition, Gerber reviews the implications of using mashups and points out a number of legal issues [4]. This includes copyright misuse, trademark violations, false advertising, contract law issues, patent infringement, warranty, and the rights and privacy of individuals. The author also observes that these legal issues require consideration prior to design or implementation. These issues further motivate the need to address accountability for enterprise mashup services.

3. Formal Definition of Accountability

In [14] it is suggested that the term 'accountability is an often used word with no common definition that can be found'. The special interest group authors [14] also conduct extensive research of the literature and have provided a definition of accountability in the context of service performance, see Table 1 below.

Table 1. Accountability for Performance

Accountability refers to the obligation a person, group, or organization assumes for the execution of authority and/or the fulfillment of responsibility. This obligation includes:

- *Answering—providing an explanation or justification—for the execution of that authority and/or fulfillment of that responsibility,*
- *Reporting on the results of that execution and/or fulfillment, and*
- *Assuming liability for those results.*

We observe that this definition requires strengthening in the multi-party scenario such as mashup service environments. In addition to non-repudiation, managing trust is also important for entities to collaborate [2]. Moreover, we wish to strengthen this definition with non-repudiation and trust with one or multiple entities.

In moving towards a definition we first propose that the essence of the accountability involves four elements from an IT perspective, refer to figure 2. In the original definition of Table 1 a person, group or organisation can be translated to the concept of identity, whereas execution of authority implies the concept of role. According to Certo, responsibility is an obligation that someone "accepts" and is not allowed to delegate or pass on to someone else [15]. Accepting implies that there is some form of agreement in place. "Answering" and "reporting" relate to disclosure. Assuming liability for results requires a way to clearly demonstrate who has done what. The term assuming liability may be considered ambiguous, and considering that trust may vary considerably in a multi-party environment, this requires strengthening to remove plausible deniability, (i.e. introduce non-repudiation).

The elements of Figure 2 involve the identity of the involved party, the role the party plays, and the agreed responsibilities in the form of contract, agreement, or signed off requirements. The last element is the performance outcome; evidence of who has done what.

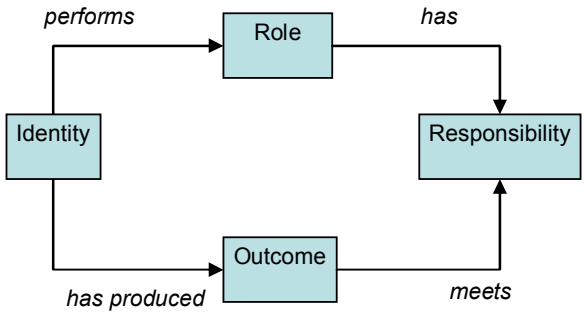

Figure 2. Accountability Elements

We now use a mashup example to demonstrate these accountability elements, see Table 2. In the scenario, Entity B offers a security trading platform to allow their customers to trade various securities globally. It has contracts with different real-time financial data providers to provide price data, which is fed into a charting provider to produce price charts. For a particular trading transaction, customer Alice initiates the trade request with Entity B. This is based on the pricing chart provided by Entity C's charting service, with real-time price input from Entity D.

Table 2. Accountability Roles in Practice

Identity	Role	Responsibility	Outcome
Alice	Trade Requestor	Enter code and bid price. Provide funds for purchase.	The request accepted by Entity B.
Entity B	Trade Provider	Display result page with data from Entity C and D. Execute trade requested. Pay Entity C and D fees due.	The trade is executed. Fund transferred from Alice's account.
Entity C	Charting Service Provider	Provide correct charted pricing indicators.	Chart is displayed and the fee is received from Entity B.
Entity D	Real-time Price Provider	Provide real-time pricing with integrity.	Data feed is provided and receive fee from B.

Using this IT services example, properties of accountability also implies:

- clear disclosure of the roles, responsibilities and transaction status by all parties;
- each party dutifully carry out their obligations;
- readily available evidence of the services rendered; and
- parties cannot repudiate services rendered.

These properties re-enforce the need for trust and non-repudiation. However, in a legal sense an automated IT system is not a legal entity that is accountable. Rather it is the person, group of people, or company which may be legally accountable. In the context of multiple entities involved in a mashup service, we now provide a more formal definition for accountability by extending the definition in [14].

In [14] responsibility is the obligation to perform, while accountability is the liability one assumes for ensuring that an obligation to perform is fulfilled [13, 14]. In addition, the term authority is the right to act without prior approval from higher management and without challenge from managing peers [13, 14]. The authors point out that authority is assigned, while responsibility is delegated. This implies a top-down decomposition of authority. Given the bottom-up method of building mashup services, this definition may not strictly apply. Rather, responsibility and authority must be sought and agreed upon between all peer content or service providers, rather than delegated. As pointed out in [15], responsibility is an obligation that is accepted; hence we observe that agreement be sought. Finally, trust may be established among peers through evidence based on historical behaviour and past interactions [2]. Considering these points we outline the extended definition, by strengthening the definition with multiparty trust and non-repudiation, making this binding to several parties; Table 3.

This definition is applicable to both the multiparty service environment (such as mashup) as well as the single party service provider. We also note that the last point of this definition uses the term trusted which also implies that all entities are authenticated. Hence, the accountable service provider would naturally maintain some form of a binding registrar that identifies the sub-ordinate accountabilities present. In order to satisfy this, the approach in [11] would seem to naturally satisfy this condition.

In light of the example and the objective to strengthen the term accountability for the broader context of multiple parties, we observe the additional properties.

- Trust: authentication of identities and agreement of accountability between all entities with evidence of behaviour; and
- Non-Repudiation: undeniable liability with full disclosure (evidence).

Table 3. Accountability for Multiple Parties

Accountability in services refers to the obligation that several persons, groups, or organizations assume for the execution and fulfillment of a service. This obligation includes:

- *Answering, providing an explanation or justification, for the execution of that authority and/or fulfilment of that responsibility;*
- *Full disclosure on the results of that execution and/or fulfillment;*
- *Undeniable liability for those result (non-repudiation); and*
- *Obtain trusted agreement of accountability from all entities involved in the service. Who in turn are bound to the obligations set out above.*

4. Service Accountability Framework

Building upon the definitions in this previous section we now propose a framework, as a meta-model and ontology, for modeling solutions in accountability for the mashup domain. The meta-model focuses upon the roles and responsibilities from an information systems perspective and is intended for IT developers. The ontology focuses upon the liabilities and agreements aspect of the definition which is useful to establish the contractual terms and definition between the respective parties.

The current literature in IT has placed much focus on the identity and performance outcome elements, which are the most difficult issues to address as that involves trust and non-repudiation. Security frameworks such as PKI alone cannot address the issues of trust in this computing environment, rather a robust security process framework and security protocol is necessary [16, 17]. As pointed out in [10], digital signatures by itself does not solve the non-repudiation issue in a multiple party environment due to the sender ambiguity problem; (i.e. a party can deny that they had received a message by blaming the non-performance of the intermediary node).

While the identity and performance outcome are important elements in the accountability framework, the role and responsibility elements are equally important. In fact, we argue that disclosure on the role and responsibility elements is the first step towards an accountability solution. This is because without a clear understanding of the roles and responsibilities by each involved party, the outcome and entity accountable can be disputed.

In a mashup service scenario, the service requester may send a request to a mashup service provider, who in turn forwards the request to the source service provider(s), before aggregating this into a new form for presentation. The issue to observe is that the original service requester may not know the identity of the original services providers. On the other hand, the original service provider is also not aware how their content may be used by the mashup service provider. This motivates the need to find an approach to enable disclosure of roles and responsibilities in mashup services, especially for the enterprise mashup services environment, that are mutually acceptable to all parties involved; whether directly or indirectly. For instance, source content provides may have restrictions on how their content may or may not be used.

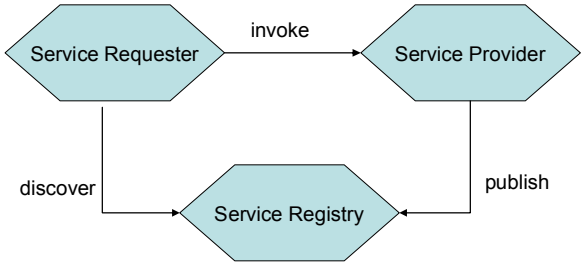

Figure 3. SOA Architectural Style and Actors

We propose a framework for modeling the behavior of mashup services based on SOA, and hence briefly visit the fundamentals of this archetype. It is commonly agreed that SOA is an architectural style that involves a triangular relationship amongst three entities: service requester, service provider and the service registry [18, 19], see Figure 3.

While the model captures the essence of the service oriented architectural model, it may fall short on enabling accountability in service oriented architectures in the mashup service environment. For instance, this does not address the roles of multiple parties and the associated responsibility of disclosure and non-repudiation.

In a mashup context, it is important to note that there are multiple service providers involved. There is also the introduction of service source as a separate entity to the provider; although, in some cases the service provider is the same entity as the service source. In practice, the service provider may engage several external content source parties to participate in constructing the service. In this situation the service provider relies upon the source for accuracies of supplied content. As suggested in [4], there are a number of legal issues that need to be considered prior to developing mashup applications. As such, both the service provider and source are required to assume responsibility to ensure the mashup service complies

with the intended application (and defined terms and conditions).

Disclosure of roles and responsibility, to a large extent, can be enabled by rich service metadata and facilitated by functions provided by the service broker; rich service metadata means adding semantics to allow machine interpretation and reasoning. Currently, the registry (UDDI) provides service metadata in terms of business entities, taxonomy and reference to service information. The registry is a dynamic name binding service that is syntax based [21]. However, in mashup several sources require identification and these may need to be trusted sources in an accountability sense. We wish to enable semantic meaning, as in OWL-S [21], and suggest a more sophisticated role to facilitate trust by enabling richer metadata to capture aspects such as traceability of service composition and responsibilities for several parties. Using this extended metadata the service request may be appropriately associated with source content that addresses both the requirement for content and the need for accountability. In some cases un-trusted content will suffice, in other situations such as enterprise mashup full accountability will be required.

4.1. Roles and Responsibilities Meta-Model

Based upon the previous discussions, we argue that the two identified roles, service requestor and service provider, do not adequately represent all the roles involved in mashup service interactions. We now propose a model to depict these relationships. The revised model is shown below, refer Figure 4. This is composed of the additional roles: service source and service broker. The service registry is still implied in this model; residing with the service broker.

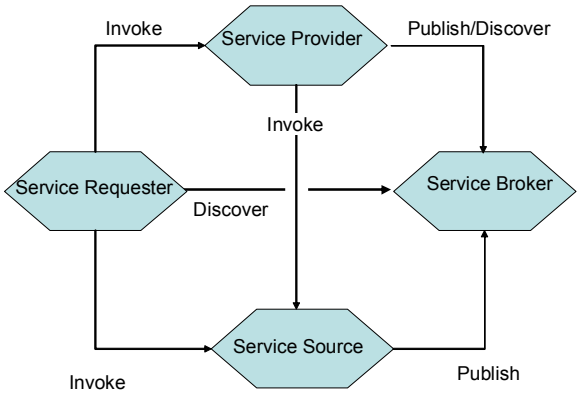

Figure 4. Mashup Role Responsibility

Note that the service provider is a special type of role in the mashup environment which plays both requester and provider at the same time. The service provider will draw upon several internal and external sources and provide a resultant mashup page to the service requester. When sourcing content from a broker or service source, the provider acts as the requestor. The service source publishes a single or discrete (common) set of content sources that may be accessed directly by the service requestor, or can be built upon and merged with other content source by a mashup service provider.

As pointed out in [4], new intermediary businesses have emerged that aggregate and broker content from several sources; in essence becoming a one stop shop of various content sources for mashup service providers. The broker supplies content to a service provider who in turn is able to mashup and repurpose the content for a service requester. This means the service requester may discover services from the broker and invoke this from a service provider. Both the service source and broker publish their available services.

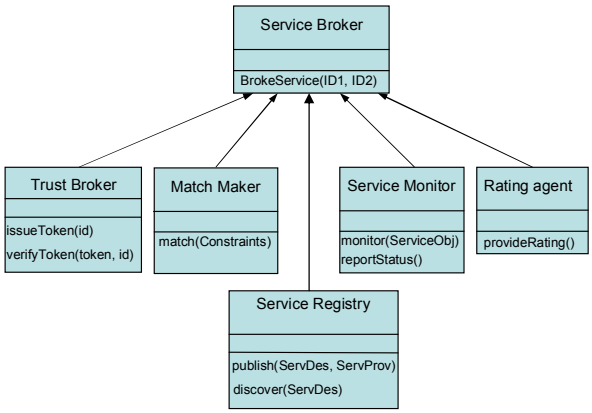

Figure 5. Expanded Roles & Responsibilities

The service broker provides several additional benefits: as a trusted brokering agent (notary for unknown sources), monitoring (audit trail and evidence) to address the disclosure and non-repudiation requirements, rating functions, and manages a combined registry and repository for multiple sources. Hence, the service broker role can be further refined into detailed roles based on these intermediary functions performed, see Figure 5. The service requester in SOA does not necessarily imply that the entity is a user. This actually refers to the client of the service, which may be another application service or software agent. In an enterprise environment, the participant role in SOA normally represents an organization or party.

The enhanced role interaction model caters for both mashup and traditional service oriented architectures. This helps to understand and define the roles and

responsibilities in service metadata. Thus the involved parties and their roles and responsibilities can be discovered and interpreted at runtime and therefore achieve the purpose of disclosure. This model is useful to information systems developers, helping them to identify roles (entities) and responsibilities in an accountable mashup services solution.

4.2. Liabilities and Obligations Ontology

The previous section focused on the roles and responsibilities in a mashup environment, outlining a model from an information systems perspective. This section models the liabilities and obligations from a legal and contractual perspective. This will assist in preparing the engagement basis and contract materials, by identifying the legal entities and artifacts that require consideration when preparing agreements to ensure accountability is established

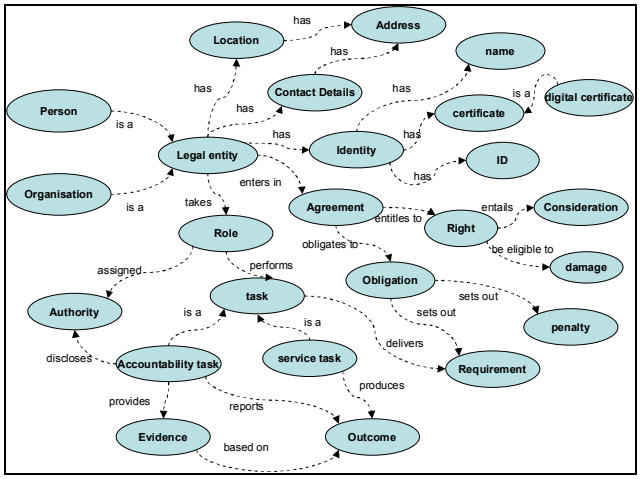

Figure 6. General Accountability Ontology

The proposed high-level accountability ontology is illustrated in figure 6. In this ontology framework, a person or organization is a legal entity that has an identity. A legal entity enters into agreement with other legal entities. The agreement embodies rights and obligations. Rights entail considerations and also imply entitlement for damage if considerations are not met. The obligation sets out the requirements that need to be delivered and penalties if the requirements are not met. Assigned with the required authority, the legal entity takes some role, which performs tasks to deliver the requirements. In the context of accountability, the task class has two subclasses, one is service task which provides the intended service; the other is the accountability task which includes disclosure of authority, outcome, and evidence.

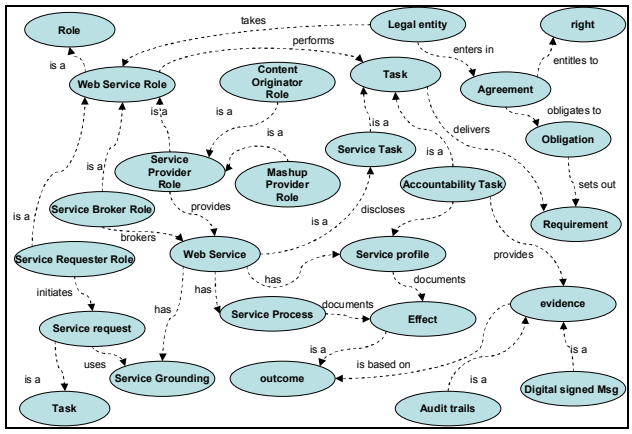

Figure 7. Mashup Accountability Ontology

OWL-S is the commonly accepted web services ontology language that provides a core set of mark up language constructs to describe web service in an unambiguous, machine interpretable form [20]. Thus it will be a natural approach to use those constructs to define the accountability elements in the service metadata. Using the general accountability model in Figure 6, we combine the high level service property constructs from OWL-S [3, 21] (service class and then its property class: service profile, service model and service grounding) to address the mashup environment. The extended accountability ontology framework is thus illustrated in Figure 7.

In the context of enterprise mashup environment, a legal entity enters into an agreement with other legal entities in order to participate in a service arrangement through web service interactions. The agreement enables the legal entity to assume a specific web service role to fulfill the obligations while receiving the considerations. As illustrated in Figure 4, the specific web service role can be service requester, service source, service provider and service broker. This role performs tasks to deliver the requirements setout within the obligations. The role performs a task which has two aspects: one is the normal web service, another is the accountability task. The accountability task includes disclosure and reporting. Disclosure in this context means disclosure of service metadata, providing evidence of the service outcome. Service metadata may include identities of the involved legal entities, roles they play, reference to the service agreement and reference to the original content in the case of mashup service. Service agreement includes terms and conditions of the service.

5. Summary and Conclusions

As mashup technology enters into the mainstream enterprise business, accountability will emerge as a key

requirement to be addressed. As pointed out in [4], a number of legal issues require consideration when using mashup solutions. We suggest that it will be increasingly important for mashup service oriented solutions to have an accountability mechanism built-in to facilitate trust and the resolution of the legal issues.

This paper builds upon existing theories by applying trust and non-repudiation in a multi-party environment for defining accountability. Using this definition we propose models that may be used by information systems developers to understand the roles and responsibilities that need to be accommodated in a mashup service solution. In addition, the liabilities and obligations are analysed. The proposed ontology helps to define the various entities and artefacts involved in a mashup service. This can be used to assist in the preparation of agreements that are required between the various entities involved in a mashup service environment. This will also help to define the scope of accountability to be addressed and will further serve to define requirements of the information systems supporting the mashup service solution in order to meet disclosure requirements.

Further Work

Several areas are suggested as further work. Firstly, one may develop an accountability assertion policy that may be included within the WS-Policy framework. Another is the inclusion of accountability metadata within the OWL-S service properties; it is suggested to use the ServiceParameter property as an extension mechanism to capture the accountability elements. Furthermore, the ontology may be refined further to establish a common accountability vocabulary useful to developers and runtime interpretation.

6. References

[1] T. O'Reilly, "What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software", O'Reilly Media Inc., September 2005.

[2] D. Huang, S. Bracher, "Towards Evidence-based Trust Brokering", First International Workshop on the Value of Security through Collaboration, Sep. 2005, pp.43–50.

[3] D. Martin, M. Burstein, *et al*., OWL-S: Semantic Markup for Web Services", W3C Member Submission, Nov. 2004.

[4] R.S. Gerber, "Mixing It up on the Web: Legal Issues Arising from Internet Mashup", Intellectual Property & Technology Law Journal, Apsen Publishers, Vol. 18(8), Aug. 2006.

[5] A. Jhingran, "Enterprise Information Mashups: Integrating Information, Simply", Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06), Seoul, Korea, Sep. 2006, pp.3-4.

[6] R. Smith, "SOA - Enterprise Mashup Services, Part 1: Real-World SOA or Web 2.0 Novelties?", SOA World Magazine, March 2007.

[7] OWL-S coalition, OWL-S 1.1 Release, 2004. Available at http://www.daml.org/services/owl-s/1.1/.

[8] R. Kailar, "Reasoning about Accountability in Protocols for Electronic Commerce", Proceedings of 1995 IEEE Symposium on Security and Privacy, IEEE Computer Society, May 1995, pp. 236.

[9] R. Kailar, "Accountability in electronic commerce protocols", IEEE Transactions on Software Engineering, IEEE Press, Vol. 22(5), May 1996, pp. 313–328.

[10] S. Bhattacharya, R. Paul, "Accountability issues in multihop message communication", 1999. Proceedings of IEEE Symposium on Application-Specific Systems and Software Engineering and Technology March 1999, pp.74–81.

[11] M.M. Tseng, J.S. Chuan, and Q.H. Ma, "Accountability Centered Approach to business process reengineering", Proceedings of the 31st Hawaii International Conference on System Sciences, Vol. 4, Jan. 1998, pp. 345 – 354.

[12] Y. Zhang, K.J. Lin, and T. Yu, "Accountability in Service-Oriented Architecture: Computing with Reasoning and Reputation", Proceedings of IEEE International Conference on e-Business Engineering, Oct. 2006, pp.123-131.

[13] B. Frost, "Measuring Performance", Fairway Press, 1998.

[14] Performance-Based Management Special Interest Group, "The Performance-Based Management Handbook: Establishing Accountability for Performance", Oak Ridge Associated Universities, Vol. 3, Sep. 2001.

[15] S.C. Certo, "Principles of Modern Management: Functions and Systems", Dubuque, Iowa: William C. Brown Publishers, 1983, p. 199.

[16] C. Ellison and B. Schneier, "Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure", Computer Security Journal, Vol. 16(1), Nov. 2000, pp. 1-7.

[17] B. Schneier, "Secrets & Lies, Digital Security in a Networked World," Wiley Publishing, Jan. 2004.

[18] Z. Stojanovic and A. Dahanayake (Eds), "Service-Oriented Software System Engineering: Challenges and Practices", IGI Global, April 2005.

[19] K. Ma, "Web Services: What's real and What's Not", IEEE IT Professional, Vol. 7(2), April 2005, pp. 14-22.

[20] D. Martin, M. Burstein, *et al*., "Describing Web Services using OWL-S and WSDL", DAML-S Coalition working document; Oct. 2003.

[21] J. Lou, B. Montrose, et al. "Adding OWL-S Support to the Existing UDDI Infrastructure", Proceedings of IEEE Conference on Web Services (ICWS'06), IEEE Computer Society, Sep. 2006, pp. 153-162.