Using Facets of Security within a Knowledge-based Framework to Broker and Manage Semantic Web Services

Randy Howard and Larry Kerschberg
George Mason University E-Center for E-Business, http://eceb.gmu.edu/
choward@gmu.edu, kersch@gmu.edu

Abstract

The concept of securing the automaton of Web services is an active research topic. We need a comprehensive and overarching framework that incorporates facets of security to broker and deliver Semantic Web Services within the context of workflow management, and addresses the issues related to Virtual Organizations. The goal is to add semantics to Web services to endow them with capabilities currently lacking in the literature, but necessary for their successful deployment in future systems.

This paper references how such a framework, called the KDSWS Framework, addresses, in a managed, integrated and end-to-end manner, how to use facets of security to broker Semantic Web Services. The following issues are addressed: semantic specification of services' security capabilities, and interoperation and evolution of the Virtual Organization.

1. Introduction

The relatively new concept of *Web services* is important to both e-Business and e-Government in that applications may exchange functionality and information over the Internet. Web services standards provide XML-based protocols to find publicly-registered services, to understand their purpose and operation, to negotiate and agree upon usage charges and Quality-of-Service commitments, and to invoke the services within the context of Internet-based workflow coordination of these services.

This paper references the Knowledge-based Dynamic Semantic Web Services (KDSWS) Framework [1] that addresses, in an integrated end-to-end manner, the life-cycle of activities involved in preparing, publishing, requesting, discovering, selecting, configuring, deploying, and delivering Semantic Web Services. In particular, the following issues are addressed: 1) semantic specification of services capabilities including quality of service, trust, and security; 2) brokering the services that are available to fulfill a request 3) workflow management; and 4) resource management, interoperation and evolution of the Virtual Organization (VO).

"Semantic Web Services" (SWS) is the term that describes our research approach. We view "Web services" as services that use little or no semantic markup, and have little of the enhanced capability described in this paper. Semantic Web technology, on the other hand adds semantic and process oriented information, together with heuristics and constraints that can be used to coordinate the activities of the VO.

Security is very important to the operation of Web services. The facets of security discussed in this paper involve properties of requests and services that support non-repudiation, authorization, authentication, trust, access control, rights and integrity. This discussion looks how these facets of security can be incorporated into the managed brokering of Semantic Web Services. Brokering, or matchmaking, involves services advertising themselves to a broker, and the broker handling queries about the available services and mediating the results for the requestor [2]. Brokering activities involve discovering available services that match the request, differentiating between the discovered services, negotiating with the top-ranked services, and finally selecting the services that best meet the request.

Section 1 has introduced the topic, while Section 2 discusses the problem space and some of the drawbacks and issues associated with existing approaches. The Knowledge-based Dynamic Semantic Web Services Framework is presented in Section 3. Section 4 presents a methodology to incorporate facets of security within the brokering activities of the KDSWS framework. Section 5 presents our conclusions and suggests areas for future research.

2. Related Work

The basic standards that support Web services are: XML-Encryption [3], WS-Security [4], XML Digital Signature [5] and Security Assertion Markup Language (SAML) [6]. Today, securing Web services involves linking many disparate protocols and technologies. Despite the fact that they are all still based on the XML

foundation, nuances of these various technologies still need to be mediated to some level to be truly interoperable. Also, these approaches do not specify aspects regarding repositories and agents.

As mentioned, Semantic Web Services attempt to address the shortcomings of Web services with respect to automation, and OWL-S is being established as a primary specification for this endeavor. However, recent research [2, 7, 8] has documented several shortcomings attributed to OWL-S.

Current research suggests syntactic, semantic and pragmatic types of brokering, or semiotic brokering to reflect all three. Syntactic brokering uses the structure or format of a task specification to match a requester with a service provider, semantic brokering uses the request's meaning and information content to match with the meaning of the offered services of the provider [9], and pragmatic[10] involves using the context of the interaction to broker services.

Current approaches to differentiate Web services utilize Service Level Agreements (SLA) [11] and Quality of Service (QoS) [12] attributes to further rank the match of the request to the providers and their services beyond service discovery. Negotiation within the Web services arena involves setting both the negotiation strategy and process via the following architectures: "fully delegated" where the negotiation service executes most of the negotiation, "interoperable messaging adapter" where the negotiation service acts a message broker with possible mapping between the messages, and "process management" where the negotiation service enforces the rules agreed by the partners while separating the decision making and from the negotiation process itself [13]. See Kim [13] for a concise summary of approaches and issues with Web services negotiation.

3. KDSWS Framework

The KDSWS Framework proposes to deliver a comprehensive and integrated end-to-end solution to dynamically broker SWS. The framework positions itself as an enterprise-scale foundation to ultimately provide VOs with the interfaces necessary to interoperate via Semantic Web Services using, and quickly adapt to, the plethora of prevailing technologies and protocols.

Like the Web Services Modeling Framework (WSMF) [14], ontologies and mediation are an integral part of this framework, and are manifested in the mapping and heuristics. The structure and design of the components target, or map to, ontological implementations such as OWL and OWL-S, albeit with potential extensions to facilitate the additional functionality. A primary function of the heuristics is to enhance the mappings ability to dynamically mediate between disparate components.

This framework has similar guiding principles, with respect to incorporating knowledge and providing a 'protocol-independent' (vs. language-independent) conceptual base, to those found in [7]. However, this framework approaches the solution spaces from a different perspective by placing more emphasis on a total enterprise solution and by providing a 'way-forward' via a guiding methodology as to how to use the structures.

As shown in Figure 1, the framework is comprised of the KDSWS Processes, KDSWS Specification, and KDSWS Functional Architecture. Note that underlines denote components specified in the diagrams and that the elements that are relevant to this discussion are highlighted in green with dashed outlines.

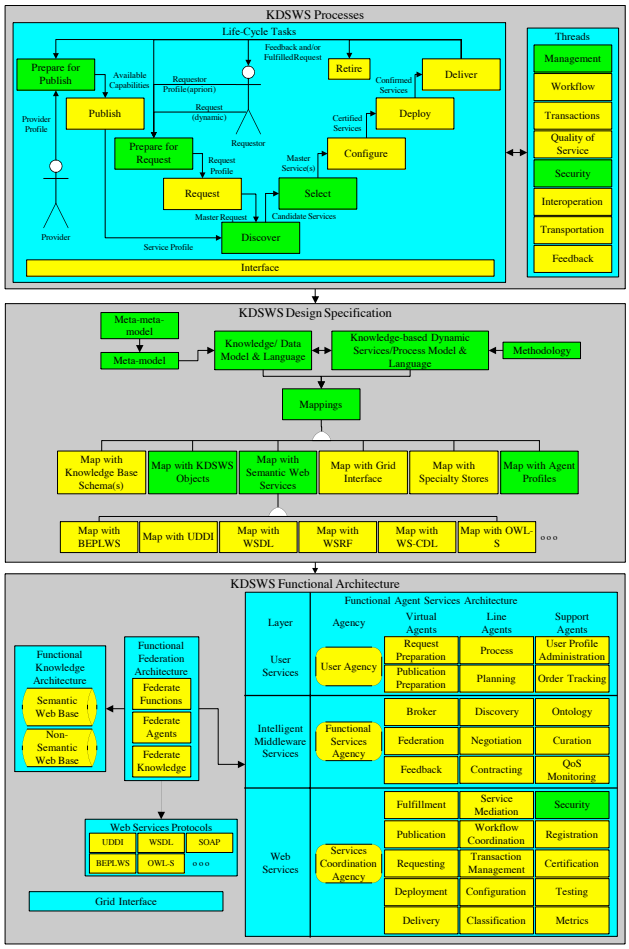

Figure 1. KDSWS Framework

The KDSWS Processes layer is segregated into Tasks and Threads; such segregation provides the ability to specify behavior based on the context of where an operation is invoked. Tasks are well-delineated steps to deliver functionality via Web services, and the brokering activities involve the Discover and Select tasks. For additional context designation, threads are defined as

layers of functionality that the tasks use to deliver the services. For this discussion, we address the issues related to Management, Workflow, and Feedback.

The KDSWS Design Specification allows the enterprise-wide data and processes needs to be stipulated for the KDSWS Processes, as well as for the backend, middleware and user services, which provides a comprehensive (by addressing backend, middleware, and user needs as well as the end-to-end issues) and integrated (by addressing both data and process needs) solution. The data aspects are captured in the meta-model and are modeled by the Knowledge/Data Model and Language (KDM/KDL) [15]. The process aspects are captured in the methodology and are modeled by the Knowledge-based Dynamic Services/Process Model and Language (KDSPM/L) [1]. This 'meta-model/process'-based approach enables an extensible framework that provides a macro-level shell to plug-in different approaches to the facets of the Web services life-cycle (e.g., negotiation, policy management, etc.) yet still provides enough structure to guide implementations.

The elements contained in the meta-model come from various sources such as the WSMF, OWL-S and Sheth [16]. The meta-model focuses on developing the 'building blocks' (i.e., constraints, preferences, profiles, capabilities, etc.) versus focusing on the higher-level and visible, finished products like a service, a profile, or request on which some approaches place their primary attention. This 'building block' approach makes the framework components composable, reusable and extensible. The meta-model distinctly defines resource-based (e.g., service, agent, protocol) and process-based (e.g., task, thread, event) classes in order to clarify the concepts involved in delivering Web services functionality.

The Mappings is a superclass of all mappings to ensure consistency across the methods to bridge the KDL/KDSPL to other technologies and standards. The mapping to other KDSWS objects (both KDL and KDSPL) enables the aggregation of the 'building blocks' to be defined by rules versus relationships. Because the rules are much easier to change, the framework is much more flexible in adapting to change. This rules-based approach facilitates the dynamic generation of the framework's profiles (e.g., requestors, providers, situations, services, etc.) in order to drive the matchmaking of a request to service(s). By combining the rules with events, the framework allows the dynamic adjustment of the process to address the 'static process declaration' issue by having certain conditions invoke specific behavior.

The profiles of the various agents are embedded throughout the KDSPL. The mapping to agent profiles identifies the points where agents are specified and organizes into a profile that contains the responsibilities of the agent within a given context. From this profile, the specification for a given agent can be engineered into a working component.

The KDSWS Functional Architecture's functional emphasis originates from the need to embed the purpose, behavior and relations within the specification via such items as the goals argument. The Functional Agent Services Architecture (FASA)'s virtual agents specify agents' aggregation and coordinate the activities of other agents. Line agents are involved in work specifications and delegate support functions to other agents (support agents). The Intelligent Middleware Services layer is supported by the Functional Services Agency whose agents include line agents that receive the task specification and plan from the User Agency, and search for appropriate Internet-based web services that can accomplish the tasks [17].

Protocols are an essential backbone for Web services that establish the expected means to communicate between end-points. There are numerous protocols (see [18]) used to support Web services, some of which are mature but most of still emerging at varying levels of maturity. Semantic expansions to these protocols will likely be necessary to take full advantage of the expanded functionality in this framework.

4. Brokering on Facets of Security

In the Prepare task, for both the request and publish stages, security paradigms, policies and capabilities are placed into catalogs, where catalogs are references for what resources are designated as 'available' at the enterprise level, local level or both.

The Publish task assigns the available security capabilities to the respective business and service information in the appropriate registries and WSDLs. The Request task selects respective capabilities as constraints and preferences for the Master Request.

The Discover and Select tasks comprise the brokering activities to match the security-related constraints and preferences such as encryption level or Single Signon support with the capabilities of the services. The Deliver task enforces and implements the security policies, as well as handling anomalies from security violations. The Retire task considers security violations as criteria for potential retirement, or at least suspended because it is not tuned with requestors.

The brokering processes in the methodology shown in Figure 2 are select portions of the KDSWS Methodology's Discover and Select tasks set within the Management, Security and Feedback Threads. The discussion below focuses on how security is incorporated in the brokering activities as presented originally in Howard [1]. Space does not permit to explain how this

methodology is specified in the KDL or KDSPL, but see [1] for more details.

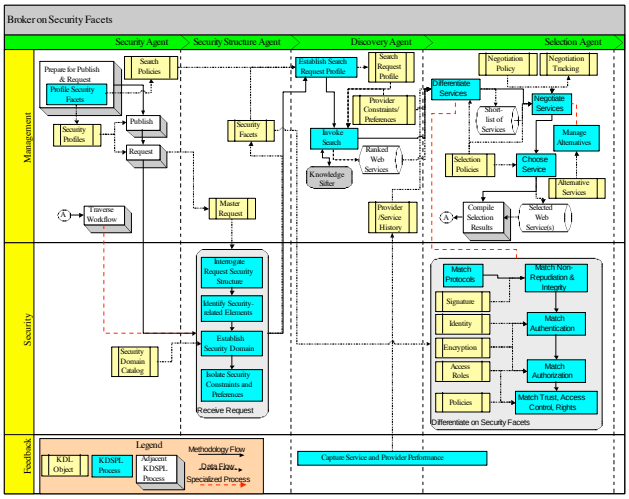

Figure 2. Brokering Methodology

The Security Agent is the agent designated as owning the security aspects of these processes. Front-end work done in the Prepare for Publish/Request phases to profile the facets improves the turnaround time of requests. Security Profiles capture such items as identity attributes (e.g., name, password, id, department, etc.) and access roles of providers, services and requestors. The Search Policies contain such information as search priorities (i.e., supporting a certain protocol base is more important than encryption method).

The Receive Request process presented here deals specifically with the facets of security discussed earlier; however, it is executed within the context that the Traverse Workflow process controls the overall flow to iterate through the services that are potentially reflected within a workflow context.

The Interrogate Request Security Structure process examines the structure of the Master Request [1] for the Identify Security-related Elements to codify the elements in the request that can be used in the brokering process. Since security profiles are very sensitive to domain, the Establish Security Domain process identifies the domain(s) to which the request belongs. The Isolate Security Constraints and Preferences process extracts the specified constraints and preferences that pertain to security out of the Master Request. The Security Agent delegates this process to the Security Structure Agent, which is a specialized support agent within the FASA.

The Security Profiles and security aspects contained in the Search Policies help prevent much of this aforementioned work from being done real-time at request receipt. Additionally, the profiles themselves can be ranked and incorporated into the search as an aggregate versus the separate components.

Next, the Discovery Agent handles the Establish Search Request Profile process which uses the Master Request and Search Policies (e.g., priorities and process) that the VO, requestor and provider wish to establish for the interaction. For example, the VO might prefer price to be deemed more important than availability, and the provider might invoke different approaches to the search depending on the level of specificity of the request. Knowledge Sifter [19] is demonstrated as the search agent selected because it handles the syntactic, semantic and pragmatic types of brokering. The Invoke Search process is started where the Search Request Profile and Provider Constraints/ Preferences provide more information to Knowledge Sifter in order for it to deliver more accurate results.

The Selection Agent rounds out the brokering activity by first invoking the Differentiate Services process to perform a finer-grain matching of the services than the search agent performed. The Capture Service and Provider Performance process feeds knowledge and activity into the Provider/Service History, which provides the SLA, reputation [12], and the activities related to security to this process. Similar to Search Policies, Selection Policies provide the priorities placed on the security parameters with regards to the selection process.

The discussion about Differentiate on Security Facets focuses on security aspects within the differentiation process. The same processing patterns in this process can be reflected in the Search Policies as well. The Match Protocols process examines the protocols that the requests use and services prefer to support as differentiation criteria. Protocols involved in this area are WS-Trust, WS-Privacy, WS-Policy, Ws-SecureConversation, WS-Federation and WS-Authorization [20]. The security-related structural elements such as the WS-Security namespaces (e.g., ds for Digital signature, wsse for WS-Security extension, wsu for Web services utility, and xenc for XML Encryption) can also be considered as criteria.

The Match Non-Repudiation & Integrity process evaluates the structure of potentially signed fragments. For example, whether signatureMethod within the Signature is DSAwithSHA1 or HMAC-SHA1 can be used as a preference or constraint. Other items such as CanonicalizationMethod (i.e., exclusive or custom), items conveyed in the reference list and *object*s held within the Digital Signature schema can be used.

The Match Authentication process focuses on the identity facet. Here, the 'strength' of the identity (i.e., the quantity of properties used to identify the resource) is used as criteria.

Encryption is evaluated in the Match Authorization process because encryption allows authorized participants to look at information, while it is intended to prevent unauthorized access. Properties such as security tokens used (i.e., Kerboros or x.509/PKI) and Encryption

Method (i.e., Triple-DES, AES-128, AES-256, AES-192) are used as parameters. The encryption of attachments, or not, can also be incorporated as attachments.

The Match Trust, Access Control, Rights process evaluates such items as trust assertions, which also deals with authentication, authorization and attributes (i.e., attributes involved with authentication and authorization) and role-based access control and rights.

This process deals with the elements of the many emerging protocols such as eXtensible Access Control Markup Language (XACML) layers and eXtensible Rights Markup Language (XrML) constructs (e.g., principal, right, resource and condition) [20].

The Negotiate Services process uses the Negotiation Policy (e.g., negotiation strategy, rules and parameters) and Negotiation Tracking (e.g., previous negotiation activity and progress within current negotiation process). This process involves a buyer, seller, and a potential marketplace in a potentially very ornate process that is beyond this discussion. This approach also uses the 'process management' type of negotiation as suggested in [13]; however, the approach can be altered as necessary. A particular facet that is yet to be worked out is the timing issues between reaching a negotiated agreement in this brokering activity and when the workflow is actually executed.

The Choose Service dynamically applies the selection priorities within the Selection Policies in order to determine the Web service that best matches the request. The Manage Alternatives process continually monitors for potential options in the case that profiled services may not meet the request exactly, but may be acceptable. Because of the workflow needs to be traversed, the interim Selected Web Service(s) need to be collected and stored via the Compile Selection Results process. Here the potentially heterogeneous results are also mediated with the overall activity into a consistent and usable form by downstream configuration activities. Meanwhile, the flow returns to deal with services potential still left to be brokered within the workflow.

Turning to specifications within the KDSWS Framework, assertions can be coded into the Constraints and Heuristics parameters (see Potter [15]) , and then mapped to SAML (see Howard [1]). Visibility within the KDSWS Specifications indicates whether access is granted to the public, to partners, or internal use only. This is contrasted with the Visibility assertion (i.e., items are in the clear or need to be decrypted) within WS-Security. The ontology within the KDSWS Framework de-conflicts these terms. Additionally, Applicability-Level specifies whether security elements pertain to the methodology, globally to the enterprise, or locally to agencies. Distribution dictates when and where to distribute policies to the enterprise.

5. Conclusions

In order to accelerate the introduction of new concepts and systems, we need approaches which automate the entire Web services life-cycle and address the issues related to Virtual Organizations. This paper presents an agent-based approach to manage the brokering of Semantic Web Services, focusing on using facets of security, within a Virtual Organization. It is part of a larger methodology, called the KDSWS Framework. The framework provides a formal model-based approach to implementing Web services that specifies the modeling, specification, design, implementation and deployment of systems composed of SWS. The goal of the framework is to support the automatic discovery, composition, execution and management of Web services for the Virtual Organization in a protocol-independent manner.

The framework provides a comprehensive solution because it addresses the backend specification for federating enterprise resources, agents and knowledge repositories. It is integrated because the data and process specifications are created using the same foundation - the KDM/KDL. Interoperability is facilitated by the integrated design space and mediation structures. The framework is knowledge-based because it uses heuristics to associate the knowledge to objects and services, and captures usage context as well. This rules-based approach facilitates the dynamic profiling of resources as well as quick adaptation to the rapidly changing Web services standard and protocol base. The term 'semantic' denotes the knowledge-based semantic specification of the relevant features and functions provided by the Web Service.

In order to address the multitude of issues in this area, we propose the KDSWS Framework as a way to combine three important, and inter-related, viewpoints:

- The KDSWS Process viewpoint addresses issues related to workflow, transaction-control, security, and interoperation in the form of "threads". The delineation between the resource-based and process-based classes in the meta-model provides crisper concepts for the framework to specify tighter designs.
- The KDSWS Design Specification viewpoint models objects, relationships, constraints, heuristics, and processes using the KDM/KDL and extensions KDSPM/KDSPL to handle special features of Semantic Web Service specifications. The advantage provided by a separate and integrated specification is the VO can adjust to a constantly changing protocol base more rapidly by developing the mappings from the consistent base versus having to recode the affected interfaces.
- The KDSWS Functional Architecture provides an agent-based architecture to implement systems via composable Semantic Web Services. The architecture

includes Functional Architectures for knowledge represented in repositories, federation policy, rules, agents, Web service protocols, and agent services to manage various aspects of deploying Semantic Web Services.

This research is still at an abstract level, and an implementation of the framework is planned where the facets of the research can be matured. Future work for the framework includes creating a modeling facility for the unique approach of the methodology that integrates the meta-model.

Finally, our research indicates that the KDSWS semantic modeling techniques and methodology, when applied to service-oriented systems exemplified by Semantic Web Services, helps to address the plethora of issues needed to successfully deploy them in real-world applications. The semantically-enabled workflow and feedback processes provide a managed approach to the dynamic delivery of Web services. The multiple viewpoints help to isolate and identify important issues and the mappings from viewpoint to viewpoint assure that the structures, operations, and constraints are properly mapped and preserved.

Acknowledgements

This work was sponsored by a NURI from the National Geospatial-Intelligence Agency (NGA). This work was also supported in part by the Advanced Research and Development Activity (ARDA).

6. References

[1]R. Howard and L. Kerschberg, "A Knowledge-based Framework for Dynamic Semantic Web Services Brokering and Management," presented at Database and Expert Systems Applications (DEXA) 2004, Accepted for publication, Zarogoza, Spain, 2004

[2]M. Paolucci, J. Soudry, N. Srinivasan, and K. Sycara, "A Broker for OWL-S Web services," presented at 2004 AAAI Spring Symposium Series, Stanford University Palo Alto, CA, 2004, http://www.daml.ecs.soton.ac.uk/SSS-SWS04/40.pdf.

[3]W3C, "XML Encryption WG," 2003, http://www.w3.org/Encryption/2001/.

[4]IBM, "Specification: Web Services Security (WS-Security)," 2002, ftp://www6.software.ibm.com/software/developer/library/ws-secure.pdf.

[5]W3C, "XML Signature WG," 2003, http://www.w3.org/Signature/.

[6]OASIS, "Security Assertion Markup Language (SAML)," 2002, http://www.oasis-open.org/committees/security.

[7]A. Gomez-Perez, R. Gonzalez-Cabero, and M. Lama, "A Framework for Design and Composition of Semantic Web Services," presented at 2004 AAAI Spring Symposium Series, Stanford University Palo Alto, CA, 2004, http://www.daml.ecs.soton.ac.uk/SSS-SWS04/44.pdf.

[8]P. Mika, M. Sabou, A. Gangemi, and D. Oberle, "Foundations for OWL-S: Aligning OWL-S to DOLCE," presented at 2004 AAAI Spring Symposium Series, Stanford University Palo Alto, CA, 2004, http://www.daml.ecs.soton.ac.uk/SSS-SWS04/23.pdf.

[9]M. Nodine, A. H. H. Ngu, A. Cassandra, and W. G. Bohrer, "Scalable semantic brokering over dynamic heterogeneous data sources in InfoSleuth," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 1082-1098, 2003

[10]A. de Moor and W.-J. van den Heuvel, "Web service selection in virtual communities," presented at System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, 2004

[11]A. Dan, H. Ludwig, and G. Pacifici, "Web service differentiation with service level agreements," 2003, http://www-106.ibm.com/developerworks/webservices/library/ws-slafram/.

[12]S. Kalepu, S. Krishnaswamy, and S. W. Loke, "Verity: a QoS metric for selecting web services and providers," presented at Web Information Systems Engineering Workshops, 2003. Proceedings. Fourth International Conference on, 2003

[13]J. B. Kim, A. Segev, A. Patankar, and M. G. Cho, "Web Services and BPEL4WS for Dynamic eBusiness Negotiation Processes," http://www.ieor.berkeley.edu/~jinbaek/publications/ICWS03.pdf.

[14]D. Fensel and C. Bussler, "The Web Service Modeling Framework WSMF," http://www.swsi.org/resources/wsmf-paper.pdf.

[15]W. D. Potter and L. Kerschberg, "The Knowledge/Data Model: An Integrated Approach to Modeling Knowledge and Data," in *Data and Knowledge (DS-2)*, R. A. Meersman and A. C. Sernadas, Eds.: Amsterdam: North Holland, 1988

[16]A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, and Y. Warke, "Managing semantic content for the Web," *IEEE Internet Computing*, vol. 6, pp. 80-87, 2002

[17]L. Kerschberg, "Functional Approach to in Internet-Based Applications: Enabling the Semantic Web, E-Business, Web Services and Agent-Based Knowledge Management," in *The Functional Approach to Data Management*, A. Poulovassilis, Ed. Heidelberg: Springer, 2003, pp. 369-392

[18]L. Wilkes, "The Web Services Protocol Stack," CBDI Web Services Roadmap, 2004, http://roadmap.cbdiforum.com/reports/protocols/index.php.

[19]L. Kerschberg, M. Chowdhury, A. Damiano, H. Jeong, S. Mitchell, J. Si, and S. Smith, "Knowledge Sifter: Ontology-Driven Search over Heterogeneous Databases," presented at SSDBM 2004, International Conference on Scientific and Statistical Database Management, Santorini Island, Greece, 2004

[20]J. Rosenberg and D. Remy, *Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. Indianapolis, Indiana: Sams, 2004