Security Attack Ontology for Web Services

Artem Vorobiev and Jun Han
Faculty of ICT, Swinburne University of Technology, Melbourne, Australia
{avorobiev, jhan}@ict.swin.edu.au

Abstract

Web services (WS) have become a significant part of the Web because of such attractive features as simple to use, platform independence, and XML/SOAP support. However, these features make WS vulnerable to many new and inherited old security threats. Semantic WS, which are capable of publishing semantic data about their functional and non-functional properties, add even more security issues. Now, it becomes easier to attack WS because their semantic data is publicly available. To register and prevent these attacks, especially distributed attacks, new distributed firewalls and intrusion detection systems (F/IDS) have to be applied. However, these F/IDS can be developed by different vendors and they do not have the way to cooperate with each other. This problem can be solved if various F/IDS share a common vocabulary, which can be based on ontologies, to allow them to interact with each other. In this paper, we describe WS security threats and state that they have to be analysed and classified systematically in order to allow the development of better distributed defensive mechanisms for WS using F/IDS. We choose ontologies and OWL/OWL-S over taxonomies because ontologies allow different parties to evolve and share a common understanding of information which can be reasoned and analysed automatically. We develop the security attack ontology for WS and illustrate the benefits of using it with an example.

1. Introduction

Web services (WS) have been deployed by many companies recently including governments, banks, large corporations etc because of their simplicity of use, platform independence, XML/SOAP support and rich functionality and interoperability. However, WS also have raised many new unexplored security issues as well as new ways of exploiting inherited old security threats. Semantic WS, which can publish the information about their functional and non-functional properties, add additional security threats. Attackers do not need to scan the Web in order to find targets. They just go to UDDI Business Registry (UBR) and get all the information they need to attack WS. Currently, there are several servers that are supported by Microsoft, IBM, SAP, etc. They have user friendly interfaces with rich functionality that can be utilized as a perfect tool for the first stage of an attack. Actually, the whole WS attack tree consists of several stages during which an attacker discovers weakness, then penetrates the WS layer and gets access to mission critical applications and infrastructures.

For example, the XML Injection attack [20], which is a new emerging attack class, occurs when user input is passed to the XML stream. This attack can be stoped by scanning the XML stream. Another type of attacks on WS is Denial of Service (DoS) attacks when hackers can send extremely complicated but legal XML documents. It forces the system to create huge objects in memory and deplete system's free memory. Distributed and multi-phased attacks such as the Mitnick attack [22] are even more dangerous because intrusion detection systems (IDS) [1,15] can detect them only by acting as a coalition. Other attacks are old attacks (OWASP Tot 10) [20] that "shine" in the WS environment.

To discover and resist these attacks, especially distributed and multi-phased attacks, new distributed firewalls/IDS (F/IDS) have to be utilized. However, because of interoperability problems (F/IDS can be developed by different vendors) it can be very difficult to create the distributed F/IDS system. The easiest solution in terms of time, money and efficiency is to develop the common vocabulary for F/IDS, which is based on ontologies and allows various F/IDS to interact with each other. Ontologies are chosen over taxonomies because of their capability of publishing semantic data. They allow different parties to share a common understanding of information that can be analysed and reasoned automatically. Also ontologies can evolve in time. For example, if one of F/IDS from a coalition of F/IDS detects a new attack, it adds the

Proceedings of the Second International
Conference on Semantics, Knowledge, and Grid (SKG'06)
0-7695-2673-X/06 $20.00 © 2006 IEEE

attack as a new class to the security attack ontology and shares the ontology with other members of a coalition. Our security attack ontology can be represented in OWL/OWL-S [18,19], however, due to space limitation, it is not defined in OWL/OWL-S syntax in this paper.

The paper is organised as follows. In section 2, we introduce an example and identify the need to consider the security attack ontology for WS. In section 3, we present WS security threats and present our ontology. In section 4, we illustrate the use of our ontology through an example. In section 5, we describe related works. We conclude and identify future work in section 6.

2. Background and motivation

2.1. An example

The Mitnick attack is related to the men-in-the-middle (MITM) attack and exploits weakness of the design of TCP protocol in making a TCP connection between two hosts (a three way handshake): 1. Host 1 (**H1)** initiates a TCP connection with Host 2 (**H2)** by sending a Syn packet to **H2**, 2. **H2** sends a Syn/Ack packet to **H1** in order to establish a TCP connection with **H1**, 3. **H1** receives a Syn/Ack packet and sends an acknowledgment on getting it by sending a Syn/Ack packet to **H2**. At this stage a connection is established and the handshake is completed.

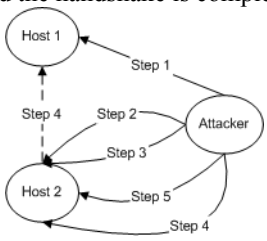

Figure 1. The Mitnick attack.

Since the Mitnick attack is a multi-phased distributed attack, it can be detected only by a coalition of Firewalls/Intrusion Detection Systems (F/IDS) distributed over the Web. The attack consists of several steps, as illustrated in Figure 1. The attacker (A) tries to attack Host 2 (H2) that trusts Host 1 (H1) using a Syn/Flood attack based upon the three way handshake during initiating a TCP connection. The attack is specified as follows (Figure 1):

1. For blocking communications between **H1** and **H2**, **A** starts a Syn/Flood attack against **H1**.
2. **A** sends multiple TCP packets to **H2** in order to predict a TCP sequence number generated by **H2**.
3. **A** pretends to be **H1** by spoofing **H1's** IP address and tries to establish a TCP session between **H1** and **H2** by sending a Syn packet to **H2** (Step 1 of a three way handshake).
4. **H2** responds to **H1** with a Syn/Ack packet (Step 2 of a three way handshake), however, **H1** cannot send a RST packet to terminate a connection because of a Syn/Flood DoS attack from Step 1.
5. **A** cannot see a Syn/Ack packet from Step 4, however, **A** can apply a TCP sequence number from Step 2 and **H1's** IP address and send a Syn/Ack packet with a predicted number in response to a Syn/Ack packet sent to **H1** (Step 3 of a three way handshake).

Now, **H2** thinks that a TCP session is established with trusted **H1**. **A** has a one way session with **H2** and can try to hijack it. It should be mentioned, that H1's IDS can register a short Syn/Flood DoS attack, while H2's F/IDS can detect an attempt to predict a TCP sequence number. If these F/IDS do not act as a coalition, they will not be able to discover the Mitnick attack.

2.2 Motivation

As it has been mentioned above, the Mitnick attack can be detected by using a coalition of distributed F/IDS. The problem in WS context is that WS and F/IDS, which should protect WS, can be distributed over the Web, controlled by different parties, and developed by different vendors. To identify such multi-phased and distributed attacks, various F/IDS from different vendors should have a common vocabulary of attacks to communicate with each other.

Firstly, to detect multi-phased and distributed attacks against WS, F/IDS should act as a coalition of F/IDS. Secondly, for better attack detection, F/IDS should be the integrated parts of the WS based system and should communicate with each other by using the same standards (XML, SOAP, OWL/OWL-S etc) as the whole system. Finally, the approach should have rich functionality and interoperability, should allow a coalition of F/IDS to evolve, should be cheap in terms of money and time consumption to implement, deploy and support.

3. WS security threats and attack ontology

In this section, we describe some of WS security threats as well as our security attack ontology and explain how they relate to each other.

3.1. Attacks on Web services

Proceedings of the Second International
Conference on Semantics, Knowledge, and Grid (SKG'06)
0-7695-2673-X/06 $20.00 © 2006 IEEE

WS have been designed to allow various technologies to be utilized at the WS layer. We specify five attack zones for this layer, as illustrated in Figure 2: Application, SOAP attacks zone, XML, Discovery, and Semantic attacks zone. We consider the application zone because many application security threats have been "reborn" in WS context. The semantic zone describes new types of attacks on semantic WS. For example, an attacker can create an ontology that can crash an OWL parser. Examples of Discovery, SOAP and XML attacks are UDDI attacks, SOAP Replay Attacks and XML Injection. Some types of attacks can be parts of different zones.

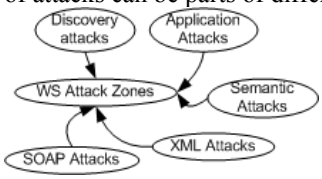

Figure 2. Attack Zones

The first step for the attacker after finding the target WS using UDDI Business Registry (UBR) is to discover points of weakness in WSDL documents which can be used as a vulnerability guide book for getting access to mission critical applications and infrastructures. In the following subsections, we discuss various attacks which can be used in different zones.

3.2. Probing attacks

The class of Discovery Attacks is represented by two subclasses: WS Probing Attacks and UDDI Attacks, as shown in Figure 3. WS Probing attacks can be subdivided into two subclasses: WSDL Scanning and Parameter Tampering.

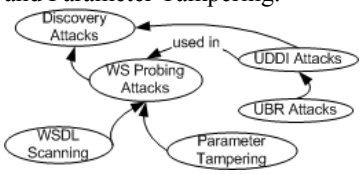

Figure 3. Class Probing Attacks

Because WSDL documents contain all functions that are available to consumers, they become an easy target for attackers. WSDL documents also have information about function parameters, hence, attackers can try to submit various parameters such as special characters in order to crash the implementation of WS. UDDI and UBR are used to find WS, however, they can also point to targets and provide all information needed to attack WS. Attackers do not need to scan the Web in order to find vulnerable WS, they just go to any UBR and find targets. Such UDDI and UBR attacks are rather dangerous because of difficulties to detect them.

3.3. CDATA Field Attacks

CDATA field allows including non-legal characters in XML documents. It can lead to several attacks including XML/XPath/SQL Injection attacks and Cross Site Scripting (XSS) attacks, as illustrated in Figure 4.

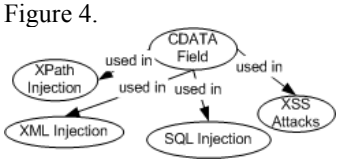

Figure 4. CDATA Field Attacks

The example below demonstrates how the script language can be used for malicious purposes:

```
<TEST>
<![CDATA[<]]>SCRIPT<![CDATA[>]]>
alert('u r hacked')
<![CDATA[<]]>/SCRIPT<![CDATA[>]]>
</TEST>
```

XML parsers usually strip "<" or ">" symbols, however, CDATA field allows to include such characters. For example, the result of parsing of this example will be:

```
<TEST><SCRIPT>alert('u r hacked')</SCRIPT></TEST>
```

The customer will see the message "u r hacked". Script languages can be rather dangerous in such situations.

3.4. WS DoS attacks

The class of WS General DoS (Denial of Service) attacks is illustrated in Figure 5. It consists of several subclasses that are introduced below.

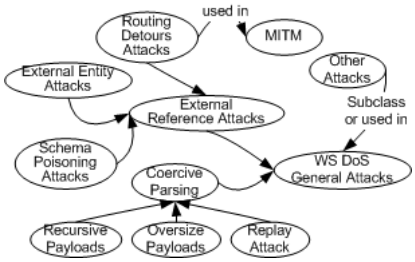

Figure 5. Class WS DoS General Attacks

The basic idea of Coercive Parsing Attacks (subclass of DoS attacks) is to exploit XML-based parts of WS (SOAP or OWL documents). Subclasses of Coercive Parsing Attacks are Replay Attacks, Recursive Payloads Attacks, and Oversize Payloads Attacks. During Replay Attacks, an attacker sends repetitive SOAP messages in order to overwhelm WS. This type of attacks is rather difficult to detect since HTTP requests are well formed and IP addresses and

network packets are valid. The capability of XML of nesting of elements within documents leads to Recursive Payloads Attacks. An attacker simply can create a document with a huge number of nested elements (more than 10000 levels) deep in order to break XML parsing. Oversize Payloads Attacks happen when an attacker creates a large XML-based document and there is not enough memory for a XML parser to process it.

The class of External Reference Attacks is another subclass of WS DoS attacks that can be subdivided into External Entity Attacks, Schema Poisoning Attacks and Routing Detours Attacks. Due to space limitation, these attacks are not presented here, however, their descriptions can be found in [13,16].

3.5. Application attacks

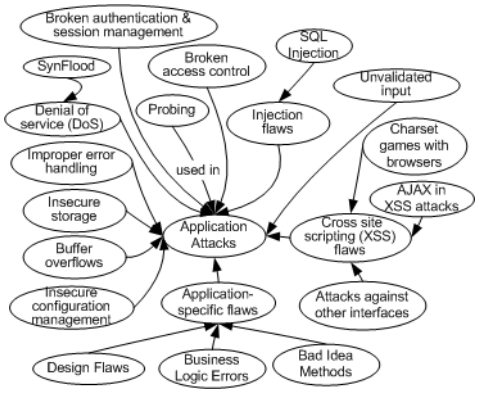

Figure 6. Application Attacks

Some of traditional application attacks are illustrated in Figure 6. For example, the SQL Injection attack, which is similar to the XML or XPath Injection attacks, can be launched if an attacker executes multiple instructions for a database using SQL separators or pipes. Detailed descriptions of other application attacks can be found in [15,20,16,13].

3.6. SOAP attacks

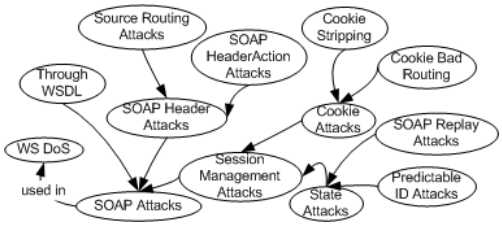

Figure 7. SOAP Attacks

The class of SOAP Attacks is shown in Figure 7. For example, SOAP Header Attacks, which can be used in WS DoS Attacks, can be realized if an attacker creates SOAP messages with very complex SOAP headers. Another example of SOAP Attacks (also used in WS DoS) is SOAP Replay Attacks which can happen if an attacker sends repetitive SOAP messages in order to overburden WS. The detailed description of SOAP attacks can be found in [5,20].

3.7. XML attacks

The class of XML Attacks, as illustrated in Figure 8, has three subclasses including Parsing Attacks, XML Injection and XPath Injection Attacks.

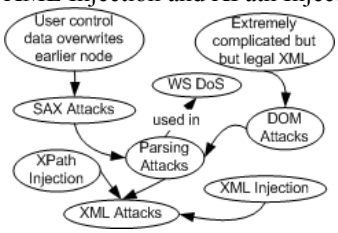

Figure 8. XML Attacks

For example, XML Injection Attacks occur when user input is passed to the XML stream. Because the XML document can be parsed by the second-tier application or the database (DB), XML code can be injected to DB, and when it is retrieved from DB, it becomes the part of the stream. The XML Injection attack tree looks as follows: 1. An Attacker Navigates to UBR and asks for a site, 2. Attaches to UDDI and asks for WS and its WSDL documents, 3. Examines them and finds dangerous methods, 4. Tests methods in order to find possibilities for the attack, 5. Changes his/her user ID in order to get control over WS. The example below demonstrates XML Injection.

```
<User>
<ID>12345</ID>
<Name>Bad Guy</Name>
<Email>badguy@oops.com</Email>
<Addr>Bad St</Addr><ID>0</ID><Addr>Bad St</Addr>
<Zip>9876</Zip>
<PhoneNumber>12345678</PhoneNumber>
</User>
```

The result of parsing by SAX parsers is ID=0. In the beginning a unique user ID equals 12345, however, an attacker enters also a fake street address (Bad St</Addr><ID>0</ID><Addr>Bad St). After parsing such address, user ID is rewritten (ID=0) because SAX parsers allow overwriting earlier nodes (SAX Attacks). DOM parsers are more complicated and intelligent and can withstand XML Injection attacks, however, they cannot resist against other types of attacks including DoS attacks when hackers can send extremely complicated but legal XML documents (DOM Attacks). It forces the system to create huge objects in memory and deplete free memory. XPath Injection Attacks [20] are similar to XML Injection Attacks.

3.8. Semantic WS attacks

Semantic Attacks, as illustrated in Figure 9, is a new emerging class of attacks that are based upon utilizing the structure or processing rules of SOAP, WS-* or semantic web standards including OWL/OWL-S, SWSL etc.

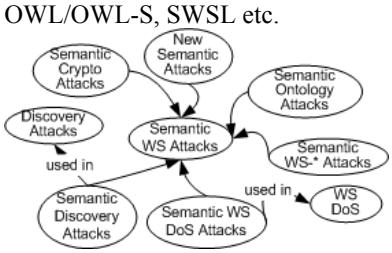

Figure 9. Semantic WS Attacks

Semantic WS DoS Attacks can happen if an attacker creates big or very complicated ontologies or schemas in order to hang a parser. Semantic Discovery Attacks can lead to information leaks because semantic data can expose a lot of information about systems infrastructure and policies (especially security policies). Semantic Crypto Attacks such as attacks on signatures may allow bogus SOAP bodies to be signed by valid signatures or bogus signatures can be utilized to sign valid SOAP bodies. Semantic WS-* Attacks will lead to many new attacks including attacks on WS-Security, WS-Policy, WS-Trust, WS-Addressing etc that will raise many security issues. Semantic Ontology Attacks will raise even more security issues because the Web 2 [17] will be widely based upon semantic web standards. It will not take much time for emerging new types of semantic attacks against WS.

4. An example

The Mitnick attack, described in Section 2.1, can be modified for using in conjunction with the XML Injection attack, introduced in Section 3.7, against WS. The attack tree is organised as follows:

1. An attacker (**A**) navigates to UBR and asks for a site.
2. **A** attaches to UDDI and asks for WSDL files.
3. Steps 3-7 are similar to Steps 1-5 from Section 2.1.
8. Now, a Host 2 **(H2)** thinks that a TCP session is established with a trusted Host 1 **(H1)**. **A** can attack **H2's** WS that believes that has a session with **H2**.
9. **A** inspects **H2's** WSDL files in order to find dangerous methods.
10. **A** tests these methods in order to find possibilities for the XML Injection attack.
11. **A** applies XML Injection for changing A's ID and getting privileges.
12. If the XML Injection attack is not successful **A** can try the SQL Injection attack or any other injection attacks against WS because H2 still believes that it is connected to H1.

To detect the modified Mitnick attack, H1's and H2's F/IDS should operate as a coalition. If H1's F/IDS detects a Syn/Flood attack, it sends a description of an attack using the security attack ontology to H2's F/IDS. Now, H2 knows that H1 is attacked and cannot be trusted until H1's F/IDS sends a confirmation message to H2's F/IDS that an attack is finished. If H1's F/IDS detects a new type of attacks it inserts a new attack class into the security attack ontology, updates it and exchanges it with H2's F/IDS. The OWL class for the modified Mitnick attack is shown as follows:

```
<owl:Class rdf:ID="&WSAttacks;WSMittnick">
 <owl:intersectionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#SynFlood"/>
  <owl:Class rdf:about="#WSProbing"/>
  <owl:Class rdf:about="#Probing"/>
  <owl:Class rdf:about="#XMLInjection"/>
 </owl:intersectionOf>
</owl:Class>
```

The instances of Syn/Flood, WS Probing, Probing and XML Injection attacks also can be defined in OWL. The ontology can evolve in time. It is also rather easy and cheap in terms of time and time consumption to implement and deploy the approach. F/IDS from different vendors just need to support ontologies, semantic web and WS standards.

5. Related work

Currently, there are several WS-* standards [7,4] that can help to protect WS including WS-Security, WS-Policy, WS-Trust, WS-Privacy, WS-Federation, WS-SecureConversation, WS-Authorization etc. The WS-* specifications can be used as building blocks for creating WS defence, however this task can be rather complicated. Also, current WS-* standards do not consider semantic data of WS.

There are several approaches that allow WS to express their semantic information and to use ontologies including Semantic SWSL [21], WSMO [24], KAoS [8], and METEOR-S [14]. However, these approaches are not specifically designed for specifying different security aspects of WS based systems.

Intrusion detection systems (IDSs) [1,9,6] are the second line of defence. They are subdivided into anomaly detection, which detects novel attacks, and signature detection (misuse detection) systems for detecting known attacks. The problem of

normal/abnormal behaviour is studied in [2]. To understand the methods used in IDSs, it is worth to know about Denial of Service (DoS)/Distributed DoS (DDoS) attacks and their classifications [15].

An ontology for creating distributed defences using IDS is introduced in [22], however, it takes into account only application attacks. Some security ontologies for WS are described in [3,12]. [12] is partially based on [3] and describes types of security information including security mechanisms, protocols, algorithms, objectives, and credentials using OWL-S [19]. It is applied to SOA to show how WS can publish their security requirements and capabilities. Also, security requirements and capabilities as well as security properties of WS and security policies of WS based systems can be expressed in SCL/E-SCL [10,11,23] which allows automatic reasoning.

Many security threats of WS as well as attacks against them and defence techniques are presented in [20,16,13,5]. However, these works are not well classified, do not describe semantic attacks against WS and do not use ontologies.

6. Conclusion and future work

Web Services (WS) are loosely-coupled components, which can create flexible, platform independent and scalable architectures with rich functionality and support of various WS-* standards. However, WS have introduced many new security threats and new types of attacks. To deal with them, it is not enough to have firewalls/intrusion detection systems (F/IDS) on a particular host. F/IDS from different software vendors should be distributed over the Web and should have the common vocabulary that provides the basis for F/IDS cooperation and evolution. In this paper, we have presented our approach to describing such vocabulary that is based on ontologies which can be defined in OWL/OWL-S. Our security attack ontology, which is easy to implement and deploy, can evolve and allows different F/IDS to interoperate with each other in order to protect WS based system from different types of attacks including distributed and multi-phased attacks. In future work, we will create the security defence ontology and combine it with our security attack ontology and approaches from [10,11,23] in order to develop better distributed defensive mechanisms for WS based systems.

7. References

[1] S. Axelsson, "Research in Intrusion-Detection Systems: A Survey", *Technical report 98–17*, Chalmers University of Technology, 1998.

[2] M. Burgers, H. Haugerud, S. Straumsnes, and T. Reitan, "Measuring System Normality", *ACM Transactions on Computer Systems*, Vol. 20, No. 2, May 2002, pp. 125–160.

[3] G. Denker, S. Nguyen, and A. Ton, "OWL-S Semantics of Security Web Services: a Case Study", *SRI International*, Menlo Park, California, USA., 2004.

[4] T. Erl, "WS-* Specifications, An Overview of the WS-Security Framework", 2004.
http://www.ws-standards.com/ws-security.asp.

[5] S. Faust, "SOAP Web Services Attacks: Are your web applications vulnerable?", *SPI Dynamics*, 2003.

[6] R. Janakiraman, M. Waldvogel, and Q. Zhang, "Indra: A peer-to-peer approach to network intrusion detection and prevention", *IEEE WETICE2003,* Austria, June 2003.

[7] IBM and Microsoft, "Security in a Web Services World: A Proposed Architecture and Roadmap", A joint security whitepaper from IBM and Microsoft, April 2002.

[8] KAoS. http://www.ihmc.us/research/projects/KAoS/.

[9] Y. Kim, W. C. Lau, M. C. Chuah, and J. H. Chao, "PacketScore: Statistical-based Overload Control against Distributed Denial-of-Service Attacks", *IEEE INFOCOM*, March 2004.

[10] K. Khan and J. Han, "A Security Characterisation Framework for Trustworthy Component Based Software Systems", *COMPSAC2003*, USA, 2003.

[11] K. Khan, "Security Characterisation and Compositional Analysis for Component-based Software Systems", *PhD thesis*, Monash University, April 2005.

[12] A. Kim, J. Luo, and M. Kang, "Security Ontology for Annotating Resources", *ODBASE 2005*, Cyprus, 2005

[13] P. Lindstrom, "Attacking and Defending Web Services", *A Spire Research Report*, January 2004.

[14] METEOR-S, http://lsdis.cs.uga.edu/projects/meteor-s/.

[15] J. Mirkovic, "D-WARD: Source-End Defence Against Distributed Denial-of-Service Attacks", *The PhD thesis*, University of California, 2003.

[16] W. Negm, "Anatomy of a Web Services Attack: A Guide to Threats and Preventative Countermeasures", 2004.

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

[18] OWL, http:w3.org/TR/owl-features/.

[19] OWL-S, http://www.daml.org/services/.

[20] A. Stamos and S. Stender, "Attacking Web Services: The Next Generation of Vulnerable Enterprise Apps", *BlackHat2005*, USA, 2005.

[21] SWSL, http://www.daml.org/services/swsl/.

[22] J. Undercoffer, A. Joshi, T. Finin, and J. Pinkston, "A target-centric ontology for intrusion detection," *Int. Joint Conference on Artificial Intelligence*, Mexico, 2004.

[23] A. Vorobiev and J. Han, "Specifying Dynamic Security Properties of Web Service Based Systems", *SKG2006*, Guilin, China, 2006. To appear.

[24] WSMO, http://www.wsmo.org/.