A Multimedia-Specific Approach to WS-Agreement

Wilfried Jouve, Julien Lancia, Charles Consel, Calton Pu

A Multimedia-Specific Approach to WS-Agreement

Wilfried Jouve
INRIA / LaBRI,
Bordeaux, France
jouve@labri.fr

Julien Lancia
Thales / LaBRI,
Bordeaux, France
lancia@labri.fr

Charles Consel
INRIA / LaBRI,
Bordeaux, France
consel@labri.fr

Calton Pu
Georgia Inst. of Tech.
Atlanta, USA
calton@cc.gatech.edu

Abstract

WS-Agreement offers a general language and protocol to establish agreements between two parties. In principle, this generality enables a wide variety of domains to be covered. Yet, agreement terms need to be developed for each domain, to allow specific requirements to be expressed. When addressing a domain, one can either invent new agreement terms, or leverage on existing approaches.

This paper proposes to extend the WS-Agreement framework to address multimedia content negotiation. This work relies on an existing protocol for multimedia negotiation, widely used in networking and telecommunications, named Session Description Protocol. This protocol is used as a framework to cover a variety of media (audio, video, images...). Besides building on a well-proven technology, our approach naturally allows the interfacing between the Web Services realm and the networking-telecommunications realm, creating a host of new usage opportunities. Our work is illustrated by two case studies: image/audio downloading and audio/video streaming.

1. Introduction

Multimedia content is becoming pervasive in Web applications and Web resources. Examples include audio *podcasting* that rapidly emerges as a new means for disseminating radio programs, video clips that are made available for sports or movie events, and voice messages that are increasingly accessible from the Web. This emerging trend is creating a need for Web services to deliver multimedia content. Doing so relies on the ability for a service provider and a service consumer to express and then agree on their multimedia capabilities and requirements.

Multimedia Quality of Service (QoS) deals with the description of multimedia content and transport parameters. For example, features of a video content include the resolution, the number of bits per pixel, the frame rate, the codecs, etc; transport parameters include the transport protocol, the port, etc. These QoS parameters are often used by existing multimedia applications, such as audio/video streaming [1] and voice over IP [2], to cope with heterogeneous device capabilities, user preferences, and multimedia content types. Existing multimedia servers and clients express their QoS parameters and rely on negotiation mechanisms to reach an agreement on the service to be delivered.

Importantly, these issues are not directly addressed by WSDL. Indeed, WSDL focuses on describing the functional interface of a web service, exposing to consumers the operations provided by the service as well as the format of the messages to be exchanged [3]. To address these issues, the web service community has proposed the notion of Service Level Agreement.

Service Level Agreement

A Service Level Agreement (SLA) is a formal contract between a service provider and a service consumer. It guarantees quantifiable service properties at specific levels.

Additionally, SLA can specify quality dimensions, to be monitored during runtime, and actions that must be taken if the contract terms are not respected by one of the parties.

WSLA [4], WSOL [4][5] and WS-Policy [6] are SLA specification frameworks whose purpose is to allow non-functional properties of Web Services to be defined. OWL-S [7], WSMO [8] and WSDL-S [9] are semantic extensions to Web Services, enhancing the expressiveness of Web Services.

However, these standards do not fulfill the requirements for expressing and negotiating multimedia QoS.

SLA in the context of multimedia

Existing approaches to SLA either complement multimedia QoS requirements (*e.g.,* network QoS) or fall short of addressing multimedia QoS concerns.

Frameworks like WSLA concentrate on the network QoS, defining response time, service availability, service reliability, etc. Although needed, these requirements do not address the QoS parameters that are specific to the multimedia domain.

Some approaches propose mechanisms that are not of direct use to multimedia applications. For example, WSOL offers a mechanism to switch between classes of services, prompting the service consumer.

Other frameworks are highly generic and require extensive work to be instantiated in a specific domain. For example, WS-agreement provides a language to specify the general format of an agreement, including constraints and guarantees. However WS-agreement does not stipulate what parameters must be negotiated between two parties.

In general, SLA approaches are mainly concerned with network constraints and issues in e-commerce applications. They are not expressive enough to fulfill requirements specific to multimedia QoS, thus requiring extensions. Moreover, a suitable negotiation process needs to be introduced for the multimedia domain.

This paper

We introduce a QoS approach in Web Services, dedicated to multimedia applications. This approach is developed as an extension to the WS-Agreement framework. Thanks to its open-endedness, the WS-Agreement specification accepts arbitrary description formats of negotiation terms and does not impose any server-side negotiation logic.

Our proposed QoS description leverages on an industrial-strength format named Session Description Protocol (SDP) [10]. This format is widely used to describe multimedia QoS in areas such as voice over IP and Video-on-Demand (VoD). It is introduced as an extension to WS-Agreement. Not only is SDP a format but it is also defined within a negotiation mechanism [11]. Because this negotiation mechanism is similar to the one proposed by the WS-Agreement specification, its integration is made possible.

The contributions of this paper include

- Introducing multimedia QoS negotiation in Web Services;
- Proposing an XML format for specifying an SDP-based multimedia QoS parameters;
- Designing and implementing an extension to the WS-Agreement framework, dedicated to multimedia QoS;
- Validating our extension on various case studies, namely, multimedia content downloading and streaming.
- Opening up Web Services on a host of new applications in areas such as telecommunications.

The rest of the paper is organized as follows: Section 2 gives an overview of our approach. Section 3 gives details about technologies at work and how they are integrated within our approach. Section 4 presents case studies, illustrating our approach. Section 5 provides a description of the implementation. Section 6 reviews some related work and Section 7 concludes.

2. Our Approach

Introducing multimedia QoS in Web Services implies having a negotiation process. This process involves two parties and leads to the creation of an agreement or contract, followed by both parties during the service delivery. In the context of a client-server model, this negotiation process must reach a compromise between server and client capabilities to share a common view of the agreement.

A negotiation process generally consists of the following steps [12] [13]: first, the two parties advertise their own view of what should be the agreement. Then, a matchmaking is performed by a particular algorithm depending on functional and/or non-functional concerns (*e.g.,* time, money, resources). Finally, when a solution is acceptable for both sides, an agreement is created between the two parties.

There are two distinct negotiation strategies: the competitive negotiation and the cooperative negotiation. The former involves a conflict of interest between the client and the server which could imply several offer/answer exchanges before reaching an agreement. In contrast, the cooperative negotiation is often carried out by a single round between clients and servers because it is an interest-based process which leads the parties to win-win solutions. An example of this type of negotiation can be found in the Session Initiation Protocol (SIP) negotiation mechanism used in the telecommunication area [2] and illustrated in Figure 1: the caller sends an invitation to the callee (an INVITE message); this invitation includes the caller's multimedia capabilities in the form of an SDP description. This description includes multimedia parameters (*e.g.,* codecs, sampling rate) and transport mechanisms (*e.g.,* transport protocol). Then, the callee sends back a SDP description, which is a subset of the

caller's capabilities (the OK message). Finally, the caller sends an acknowledgement (ACK message) to start the multimedia session matching the negotiated parameters. Our approach uses the cooperative negotiation because we target applications where any subset of capabilities common to the server and a client is acceptable.

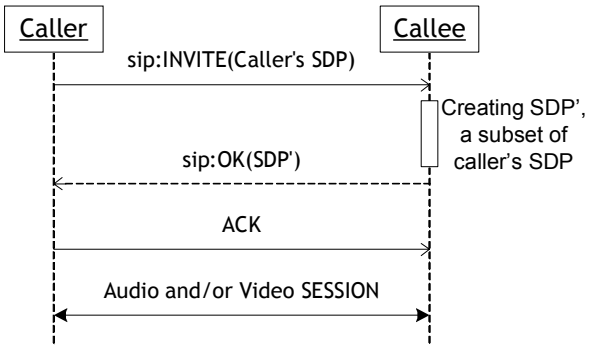

Figure 1: Negotiation with SIP

Cooperative-based negotiation and SDP form an industrial-strength basis for our approach. The scope and expressivity of this technology have been validated by multimedia QoS parameters supported by a number of deployed applications. Their standardized nature of this technology allows interoperability at a very large scale.

3. Integrating multimedia QoS in Web Services

In this section, we present both technology components that need to be integrated, namely SDP and WS-Agreement. Each technology component is introduced; technical details pertaining to their integration are described.

3.1. Session Description Protocol

SDP is a format for describing multimedia sessions in order to initiate multimedia sessions. SDP describes multimedia sessions' properties as well as characteristics of the media provided in the session. SDP captures the multimedia domain in a consistent way. Indeed, it is used in widely-spread technology components such as the Session Initiation Protocol (SIP), the Session Announcement Protocol (SAP) [14] and the Real-Time Transport Protocol (RTSP) [1].

SDP is a human-readable text protocol, which only has a descriptive role. The way SDP messages should be carried to their destination is not specified. Thus, SDP messages are generally conveyed on top of existing transport protocol such as SIP.

3.1.1. Negotiating multimedia QoS. An extension to SDP specifies a negotiation protocol that allows participants of a session to reach a mutual view of all participants' multimedia capacities. This negotiation process is based on an offer/answer model and allows communicating entities to exchange multimedia data in a compatible way.

According to the SDP offer/answer negotiating process, the negotiation initiator, called offerer, sends an SDP description containing the media and codecs it is willing to use. The other participant, called answerer, receives this offer and sends back an answer containing a matching media for each media in the offer. The answer specifies whether the stream is accepted or not, as well as IP network information used to convey the stream. Subsequently to this negotiation phase, both offerer and answerer share a mutual knowledge of each other's multimedia capacities and can thus begin a multimedia session.

3.1.2. Technical details. An SDP description consists of a session-level description followed by optionally several media-level descriptions. Descriptions are a succession of lines of text where each line contains a field identified by a unique letter and the corresponding value for this field. Session-level sections start with the 'v' field which specifies the version of the SDP protocol to be used, while each media-level description starts with a 'm' field which specifies the type of media (*e.g.,* video, audio, application) described in the section.

The session-level description fields include session name, purpose, time the session is active and optionally information about the bandwidth to be used and contact information for the person responsible for the session.

The media-level description fields include the type of media, the transport protocol to be used to convey the payload data (*e.g.,* RTP [15]), the format of the media and network information such as the remote address and ports.

SDP standard fields are often sufficient to describe a session with its comprised media. Nevertheless, SDP can be extended. Indeed, the attribute field allows specifying user-specific attributes in the session-level as well as in the media-level description, enabling SDP to be used in many potential new uses.

As multimedia formats evolve continually, SDP description's media format field structure must be readily adaptable. To reach this adaptability, audio and video content media format are often roughly described

in the media format field and completed subsequently thanks to attribute fields.

3.2. WS-Agreement

The WS-Agreement specification aims at specifying a language and a negotiation protocol to establish an agreement between a consumer and a service provider. The WS-Agreement protocol allows service providers to advertise their capabilities (generally in term of quality of service) and to provide guarantees for consumers about services they will deliver. WS-Agreement specifies XML schemas for defining the general structure of an agreement however it does not specify any format for negotiation parameters, also called agreement terms in WS-Agreement. WS-Agreement also specifies a set of port types and operations for managing the agreement life-cycle.

As shown in Figure 2, the negotiation in WS-Agreement is performed in two stages represented by a layered model: The first layer is called the agreement layer and the second one is the service layer. The agreement layer consists in creating an agreement between the service consumer and provider for a certain type of Web Services. The negotiation process involves three forms of agreements: the *agreement template*, the *agreement offer* and the (final) agreement. An agreement template consists of a context (*e.g.,* time of validity, involved parties, etc), agreement terms specific to the target domain, and constraints on these terms (e.g., maximum, minimum, enumeration, etc). There are two main types of agreement terms: *service description terms*, which define the functionality that will be delivered under an agreement, and *guarantee terms*, which specify promises and penalties if the agreement terms are not respected. Besides, *agreement constraints* specify acceptable values for agreement terms. The agreement offer is an instantiation of the agreement template by the service consumer. It specifies the requirements of the service consumer.

The agreement layer assumes that the service consumer already has the agreement template. Consequently, we have extended this layer by adding the ws:getTemplate operation which allows the service provider to advertise its agreement templates by a pull mechanism. Then, the service consumer provides values to the agreement terms according to constraints expressed by the service provider in the agreement template and makes an agreement offer. Finally, if the service provider accepts this offer, the final agreement is created and an end-point reference (EPR) is sent to the service consumer. An EPR is a reference address (compliant with the WS-Addressing specification [16]) pointing to an agreement.

The service layer represents the application-specific layer of the service being provided (*e.g.,* ws:getContent in Figure 2). When calling a service, the service consumer inserts an EPR in its request to indicate to the service provider the relevant agreement for this request. The delivered service is then adapted accordingly.

The WS-Agreement protocol is based on an offer/accept model which is quite appropriate for multimedia QoS negotiation. The cooperative negotiation only needs a single round message exchange to reach a final contract between service consumers and a service provider.

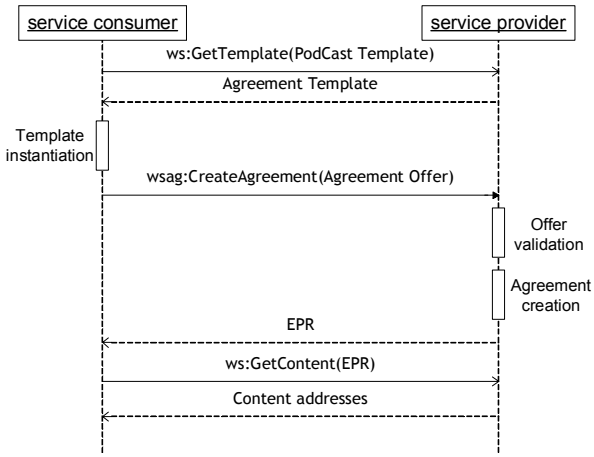

Figure 2: Negotiation with WS-Agreement

3.3. Integrating SDP into WS-Agreement

The WS-Agreement negotiation process is very close to the one used in the SDP approach. Both are based on a single round message exchange. It is referred to as the offer/answer exchange in the SDP approach, while the WS-Agreement protocol refers to it as the offer/accept exchange.

Through the service description terms, the agreement template provides the service consumer with alternatives to QoS parameters, while constraints define the server capabilities. An example of a service description term is the codec type, which can be constrained by an enumeration of codecs (*e.g.,* mp3, wma, and aac). Agreement constraints allow the content providers to advertise its capabilities. This strategy is similar to what a SIP terminal does by advertising its capabilities through the INVITE message. This “offer” step in the SDP approach is achieved by retrieving the server template in WS-Agreement.

In WS-Agreement, the “offer” step amounts for the service consumer to send an offer to the content

provider. This offer specifies the expected QoS parameters, following the constraints specified by the template. As a result, this offer contains a subset of the content provider's capabilities. This step is similar to the SDP approach (*i.e.,* the "answer") where the callee also sends a subset of the caller's capabilities.

The final step in the WS-Agreement protocol (*i.e.,* the "accept") is similar to the one in the SDP approach where the content provider or caller accepts or rejects the agreement. If it is accepted, the service is delivered following this agreement. In the same way, an SDP-based multimedia session is set up following the negotiated parameters.

The similarity between these two approaches makes it possible to integrate the SDP approach into WS-Agreement.

An XML version (based on OWL [17]) of SDP was designed to insert SDP descriptions in agreement templates. As shown in Figure 3, SDP terms are exposed as service description terms in the agreement template structure of WS-Agreement. In this example, the template is for multimedia streaming and states content provider capabilities. This content provider is able to stream either in MPV or in H261 format as specified in creation constraints of the agreement template. Some terms are chosen by the service consumer (*e.g.,* codecs in Figure 3) while others are unchangeable. The later ones are made available only to inform the service consumer which could then refuse to make an agreement offer.

```
<wsag:Template>
  <wsag:Name>StreamingTemplate</wsag:Name>
  <wsag:Context>
    [...]
  </wsag:Context>
  <wsag:Terms>
    <wsag:All>
      <wsag:ServiceDescriptionTerm>
        [...]
        <sdp:MediaAnouncement rdf:ID="MediaAnouncement">
          <sdp:MediaType>video</sdp:MediaType>
          <sdp:MediaTransportPort>0</sdp:MediaTransportPort>
          <sdp:MediaTransportProtocol>RTP/AVP</sdp:MediaTransportProtocol>
          <sdp:hasMediaFormat rdf:resource="#MediaFormat"/>
        </sdp:MediaAnouncement>
        [...]
        <sdp:MediaFormat rdf:ID="MediaFormat">
          <sdp:EncodingName>MPV</sdp:EncodingName>
          <sdp:ClockRate>90000</sdp:ClockRate>
          [...]
        </sdp:MediaFormat>
        [...]
      </wsag:ServiceDescriptionTerm>
    </wsag:All>
  </wsag:Terms>
  <wsag:CreationConstraints>
    [...]
    <wsag:Item>
      <wsag:Location>//sdp:MediaFormat/sdp:EncodingName</wsag:Location>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="MPV"/>
        <xsd:enumeration value="H261"/>
      </xsd:restriction>
    </wsag:Item>
    [...]
  </wsag:CreationConstraints>
</wsag:Template>
```

Figure 3: Agreement template for multimedia streaming

3.4. Exploring new uses

Introducing multimedia QoS in Web Services opens a great range of opportunities in the Web Services paradigm and beyond. This evolution critically relies on SDP in that it acts as a gateway between Web Services and major application areas such as telecommunications and multimedia streaming.

SIP devices have a static SDP description to carry out automatic negotiations with other SIP devices. SIP is used in Voice over IP [18] (a prime usage of SIP is 3G mobile telephony) but can also be used in ubiquitous scenarios [19] where SIP devices include webcams, TV monitors and sensors. This emerging host of SIP devices would greatly widen the scope of Web Services, if it could be operated through some interface. Such interactions could be enabled by a gateway between Web Services and SIP, as shown in Figure 4. Such a gateway is greatly simplified by our approach in that our SDP-based extension of WS-Agreement facilitates the mapping of a QoS specification from the realm of Web Services to SIP, back and forth.

With the increasing importance of multimedia streaming, Web Services would also significantly benefit from interoperability with the RTSP world. In fact, RTSP relies on the SDP format for describing multimedia streams. Such descriptions propose alternative multimedia streams, based on variations of QoS parameters, that should match user preferences and/or device capabilities. To do so, some VoD servers store variations of the same multimedia content. Alternatively, they produce tailored content on-the-fly via a transcoding server. The growing number of RTSP-based multimedia servers should further broaden the scope of Web Services, when embedding our SDP extension of WS-Agreement.

The World Wide Web is a rich source of heterogeneous multimedia content. This heterogeneity reflects the wide variety of available sources and targeted devices. A typical example of this situation is Podcasts. Podcasting is within anybody's reach from commercial radio to individuals. As a consequence, the heterogeneity of proposed content is huge. A search engine taking into account the multimedia QoS of these sources would be highly relevant. To do so, a search engine web service could enable a client to submit a QoS description prior to searching multimedia content. For a SIP device, a default description could be created based on the SDP, used in an INVITE message.

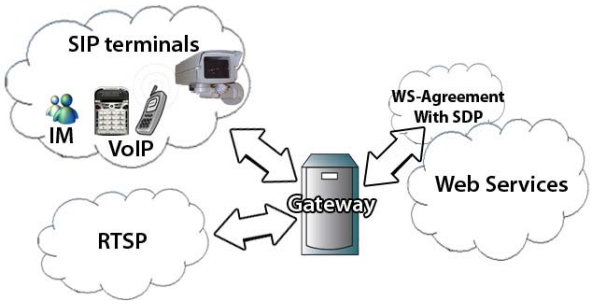

Figure 4: Potential new uses for Web Services

4. Case Studies

Multimedia-based applications differ by their transport mechanisms and by their content. As a result, they need different agreement templates and different SDP terms. Our case studies consist of two kinds of applications: content downloading and content streaming. They handle image, audio and video content. Each kind of multimedia content needs particular QoS parameters: channels for audio content, frame rate for video content, or color depth for still images. Thus, a different set of service description terms must be defined for each of these applications although some terms will be common to several applications. It will lead to different agreement templates. Because SDP originally targets streaming multimedia content, it needed to be extended to cope with multimedia QoS negotiation for still images.

We propose three case studies: an image downloading application, a podcast downloading application and a VoD application.

4.1. Server Architecture

As shown in Figure 5, our case studies can be divided in two kinds of applications: content downloading applications (including the image and podcast downloading applications) and content streaming application (*i.e.,* the VoD application). To address these applications, we have developed a server architecture that is composed of a negotiation module and a search engine module. The negotiation module implements the WS-Agreement specification extended with SDP. The search engine compares client agreements (identified by EPRs) with multimedia content available on the server-side to propose relevant content to users or clients. As our case studies focus on multimedia QoS, we implement minimal functionalities for the search engine: it only enables a search by keywords on titles and content publishers.

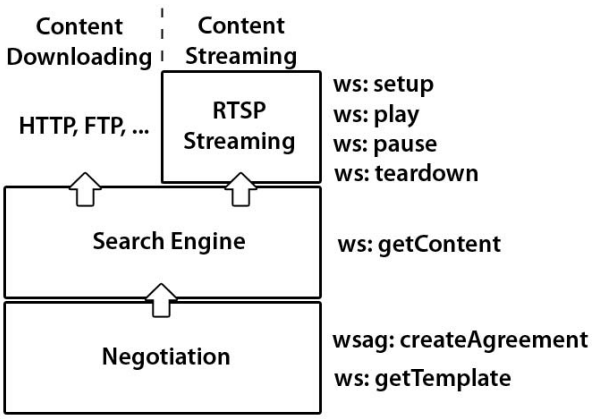

Figure 5: Server Architecture

4.2. Negotiation

The negotiation module allows clients or users to retrieve the agreement template through the ws:getTemplate operation.

The creation of the agreement offer is then performed either manually by the user or automatically by the client. Through a GUI, the user can manually fill a form generated from the template to create its agreement offer. Form fields represent service description terms and are constrained following the creation constraints of the agreement template. In this way, the user can enter the desired QoS, while respecting the server capabilities. The user can also specify mandatory service description terms (*e.g.,* if the device is a PDA whose resolution cannot exceed 600x400, no multimedia content with a resolution higher than this limit will be proposed). If the user's device already has a static SDP description, the user does not have to manually fill the template: the client can automatically create the agreement offer.

Once the client or user has created its agreement offer, it uses the wsag:createAgreement operation to generate the final agreement on the server-side. If the agreement offer is accepted, the wsag:createAgreement result is an EPR, which permits to uniquely identify the final agreement. The association between EPRs and agreements is stored at the server-side, while the client locally keeps the EPR. Then, the client calls application-specific operations (*e.g.,* search engine operations) with an EPR to personalize their behaviors.

4.3. Search Engine

The search engine uses the ws:getContent operation which needs an EPR in its input. The ws:getContent output is a list of URLs to multimedia content. These results are displayed in a web interface. Depending on

the chosen EPR, this multimedia content could be images, video or audio content. Because content downloading applications use HTTP or FTP URLs, they are directly handled by the Internet browser. In contrast, streaming applications need to both handle RTSP commands (*e.g.,* play or teardown) and start an RTP client to receive the stream.

On the server-side, multimedia content is enhanced with metadata in XML containing values for all service description terms of the associated agreement template. Each of these XML files also contains a title, a publisher and an address to a thumbnail. This additional information is used both for keyword searching and for presentation purposes when displaying results.

To select relevant content towards the client's agreement, the server performs a comparison between this agreement and the metadata of all contents by applying a particular algorithm. This algorithm rejects all contents that have a term, defined as mandatory by the client, not matching exactly the agreement. It gives a mark of 0 or 1 when evaluating string-typed terms (1 when matching). For real or integer-typed terms, the mark is computed following the difference between the agreement value and the content term value (1 when matching exactly). Based on these intermediate marks, a final mark is computed for each content and the list of ranking contents is displayed with a level of relevance. This algorithm is preliminary, others are being considered and assessed.

4.4. Multimedia Content Streaming

The multimedia content streaming module is an interface to an RTSP streaming server. It permits media to be accessed on demand, while remaining in the Web Services paradigm. It consists of four operations that mirror the main RTSP commands: Setup, Play, Pause and Teardown.

The Setup operation accepts an EPR to a previously negotiated contract and the name of the content to be streamed as input arguments. On the Web Services application server-side, this operation sends a Setup request to an RTSP server, leading the server to reserve resources needed to stream the media. Some of the parameters previously negotiated in the agreement (*e.g.,* client ports and streaming protocol) are included in this request. The response to the Setup operation is the unique identifier of the session that the client can use to execute other operations on the stream.

The Play and Pause operations accept a previously retrieved unique identifier as input. On the server-side of the Web Services application, the Play operation sends a Play request to the RTSP server, which causes the streaming server to start streaming the multimedia content to the client. The media streamed to the client respects the codec and codec parameters previously negotiated in the agreement. On the client side, provided a multimedia player is configured according to the negotiated parameters, the stream can be received. Similarly, the Pause operation stops momentarily the stream, which can be restarted later thanks to the Play operation.

Finally, the Teardown operation, which also accepts a unique identifier as input, instructs the server to stop the stream and free the associated resources. The stream cannot be restarted subsequently and the unique identifier has no longer any meaning.

5. Implementation

Well-known platforms to develop web services include IBM WebSphere, Eclipse SDK with Web Tools plugin and Microsoft .NET. We developed our prototype based on Eclipse SDK and the Web Tools plugin [22], including an implementation of the SOAP server named AXIS and combined with the Apache Tomcat application server. This is a feasibility study of our idea; its integration into a full implementation of WS-Agreement (*e.g.,* Cremona [20] in the IBM ETTK [21]) is ongoing work.

On the server side of the streaming application, we used the VLM (VideoLan Manager) extension of the VLC player that allows to stream videos on demand using the RTSP protocol. The Web Services application server connects to the RTSP port and sends RTSP commands to the server. On the client side, we used the VideoLan VLC player. We chose the VLC solution because it is an open source solution and because it supports a large panel of media formats.

The JDOM library [23] was used to manipulate XML files. The MySQL Connector/J library [24] was used to store agreements in a MySQL database on the server-side.

6. Related Work

6.1. Unifying SIP with Web Services

Chou *et al.* developed an approach to enabling telecommunication services in Web Services [31][32]. This approach also combines SIP and Web Services. It relies on the concept nodes, named WSIP nodes, which are both SIP endpoints and web service SOAP nodes. As a result, these nodes act as gateways between these two paradigms. Benefits are manifold; it allows a SOAP client, which amount to an Internet browser, to communicate with SIP terminals. As in our approach, they leverage on existing standards: the strength of SIP

is used for session signaling, while Web Services bring the client portability and their capacity of integration with business transactions. This work does not reuse existing frameworks to perform QoS negotiation whereas our approach is based on the WS-Agreement framework.

6.2. WS-Agreement Extensions

WS-Agreement is a relatively recent specification, with a formal definition [25] that extends it by expanding the set of states. This extension aims to allow renegotiation of terms when they are close to be violated. A framework and implementation [26] automates the matching of service provider and service consumer based on their agreements. This work uses OWL ontologies to capture domain-specific information. WS-Agreement has been combined with WSCL (Web Service Conversation Language) [27] to support negotiations richer than offer/accept model. Our work also extends WS-Agreement and extends it, for the unexplored domain of multimedia-specific aspects of agreement specification and negotiation.

6.3. Multimedia QoS Negotiations

General negotiation of QoS parameters for Web Services includes the CC/PP [28] framework and the QCWS [30]. CC/PP uses an RDF [29] vocabulary to specify capabilities and preferences of different parties. The negotiation model used in CC/PP is inspired by HTTP negotiations and thus it is less powerful than the one used for WS-Agreement. In contrast with our negotiation model, the CC/PP model does not lead to a shared knowledge of the capabilities and preferences present on the provider/consumer service side.

In the QCWS architecture, a QoS broker collects QoS information about the service provider. The QoS broker accepts QoS requests from clients and establishes contracts with service providers on behalf of the client. In addition to service and network performance as QoS parameters, our work adds multimedia content properties as new QoS parameters. While both CC/PP and QCWS provide different ways to describe multimedia properties as QoS parameters, our work leverages on widely accepted protocols (SDP) and provide interoperability with existing multimedia applications and an estimated billions of SIP-compatible devices.

7. Conclusion

This paper presents a novel approach to introducing negotiated multimedia QoS through Web Services. We extend the WS-Agreement specification with a widely supported (by the telecommunications industry) format and negotiation mechanism called Session Description Protocol (SDP). From the negotiation protocol point of view, our work adds expressive power to QoS negotiation protocols; for example, it supports multimedia content properties as new QoS parameters. From the systems point of view, we have implemented the new extensions on a working prototype based on Eclipse SDK. The prototype demonstrates both the feasibility and practicality of our approach.

From the deployment point of view, one of the major advantages of our approach is the support of SDP as the language describing QoS parameters. This feature allows the easy adoption of our proposal by a vast number of current and future SIP-compatible devices such as handsets and sensors. In summary, our work leverages on existing technologies, increasing the interoperability of Web Services and opening up a host of new opportunities of interactions with other areas such as telecommunications and multimedia streaming.

We did not find any fundamental roadblocks in the WS-Agreement or SDP standards that prevented the work described in this paper. Consequently, our successful integration of SDP with WS-Agreement can be seen as an encouraging sign for future work in this line of research, towards further integration of standards that deepen the semantics of web service protocols such as WS-Agreement.

8. Acknowledgement

This work has been partly supported by the *Conseil Régional d'Aquitaine* under contract 20030204003A, by NSF/CISE IIS and CNS divisions, and Hewlett-Packard.

9. References

[1] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)", RFC 2326, Apr. 1998.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, and E. Schooler. "SIP: Session Initiation Protocol" RFC 3261, Jun. 2002.

[3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (WSDL) 1.1", *World Wide Web Consortium (W3C)*, http://www.w3.org/TR/wsdl.

[4] A. Dan, R. Franck, A. Keller, R. King, and H. Ludwig, "Web service level agreement (WSLA) language specification", IBM Web Services Toolkit, Feb. 2003.

[5] V. Tosic, K. Patel, B. Pagurek, B. Esfandiari, W. Ma, "WSOL - Web Service Offerings Language", In Revised Papers From *the international Workshop on Web Services, E-Business, and the Semantic Web*, 2002.

[6] D. Box, F. Curbera, M. Hondo, C. Kaler, D. Langworthy, v Nadalin, N. Nagaratnam, M. Nottingham, von C. Riegen, and J. Shewchuk, "Web Services Policy Framework (WS-Policy)", Dec. 2002.

[7] "Web Ontology Language for Web Services", http://www.daml.org/services/owl-s.

[8] "Web Service Modeling Ontology", http://www.wsmo.org/.

[9] R. Akkiraju, J. Farrell, J.Miller, M. Nagarajan, M. Schmidt, A. Sheth, and K. Verma, "Web Service Semantics - WSDL-S", A joint UGA-IBM Technical Note, version 1.0,. Technical report, Apr. 2005.

[10] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, Apr. 1998

[11] J. Rosenberg, H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, Jun. 2002

[12] R. Huang, "Negotiation Modeling and E-shopping Agents", in the IEEE CS proceeding of the *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03)*, pp3-10, Xi'an, China, Sep. 2003.

[13] W. Picard, "NeSSy: Enabling Mass E-Negotiations of Complex Contracts", *dexa*, p. 829, *14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, 2003.

[14] M. Handley: "SAP: Session Announcement Protocol", Internet-draft, Apr. 1998.

[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications", RFC 1889, IETF, Jan. 1996.

[16] Web Services addressing (ws-addressing), W3C member submission, Aug. 2004. http://www.w3.org/Submission/ws-addressing/.

[17] S. Bechhofer, F. Harmelen, J. Hendler, D. Horrocks, I. McGuinnes, P. Patel-Schneider, and L. A. Stein, "OWL Web ontology language reference," Technical report, *World Wide Web Consortium (W3C)*, 2004.

[18] L. Burgy, C. Consel, F. Latry, L. Réveillère, and N. Palix. "Telephony over IP: Experience and Challenges", *ERCIM News*, 63:53, Oct. 2005.

[19] S. Berger, H. Schulzrinne, S. Sidiroglou, and X. Wu, "Ubiquitous computing using SIP", In Proceedings of the *13th international Workshop on Network and Operating Systems Support For Digital Audio and Video (NOSSDAV '03)*, Monterey, CA, USA, ACM Press, New York, NY, 82-89, Jun. 2003.

[20] H. Ludwig, A. Dan and R. Kearney, "Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements", In Proceedings of the *2nd International Conference on Service Oriented Computing*, 2004.

[21] "ETTK, Emerging Technologies Toolkit," IBM alpha Works, http://www.alphaworks.ibm.com/tech/ettk.

[22] Eclipse Web Tools Platform project http://www.eclipse.org/webtools/.

[23] The JDOM library, http://www.jdom.org/

[24] The MySQL Connector/J library http://www.mysql.com/products/connector/j/.

[25] M. Aiello, G. Frankova, and D. Malfatti, "What's in an Agreement? An Analysis and an Extension of WS-Agreement," In Proceedings of *the 3rd International Conference on Service Oriented Computing (ICSOC'05)*, 2005.

[26] N. Oldham, K. Verma, A. Sheth, and F. Hakimpour, "Semantic WS-agreement partner selection," In Proceedings of *the 15th international Conference on World Wide Web*, 2006.

[27] S. Paurobally, and N. R. Jennings, "Protocol engineering for web services conversations," *Int J. Engineering Applications of Artificial Intelligence*, 2005

[28] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran, "Composite Capability/ Preference Profiles (CC/PP): Structure and Vocabularies," *World Wide Web Consortium (W3C)*, http://www.w3.org/TR/CCPP-struct-vocab/, 2004.

[29] D. Brickley, and R.V. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema". *World Wide Web Consortium (W3C)*, Jan. 2003.

[30] H. G. Chen, T. Yu, and K. J. Lin, "QCWS: an implementation of QoS-capable multimedia Web services", Proceedings of the Fifth *International Symposium on Multimedia Software Engineering*, Dec. 10-12, 2003, pp.38-45.

[31] F. Liu, W. Chou, L. Li, and J. Li, "WSIP – Web Service SIP Endpoint for Converged Multimedia/Multimodal Communication over IP", Proceedings of the *IEEE International Conference on Web Services* (ICWS'04), 2004.

[32] W. Chou, L. Li, and F. Liu, "Web Service Enablement of Communication Services", Proceedings of the *IEEE International Conference on Web Services* (ICWS'05), 2005.