A Semantics- and Data-Driven SOA for Biomedical Multimedia Systems

Shih-Hsi Liu, Yu Cao, Ming Li, Pranay Kilaru, Thell Smith, Shaen Toner
Department of Computer Science
California State University, Fresno
{shliu, yucao, mingli, kilarupranay12, thell, monoman}@csufresno.edu

Abstract

Due to the problems of heterogeneous data and platforms, abundant functional and QoS requirements, high data size, and tangling correlation between data/contents and software functionalities, developing large-scale biomedical multimedia database systems is a challenging task. This paper presents a semantics- and data-driven Service-Oriented Architecture (SOA) to take the interoperability and scalability advantages of conventional SOA and solve the aforementioned problems. By establishing data ontology with respect to data properties, contents, QoS, and biomedical regulations and expanding service ontology to describe more functional and QoS specifications supported by services, appropriate services for processing biomedical multimedia data may be discovered, performed, tuned up or replaced as needed. Additionally, six transmission services are introduced to support dynamic adaptation under specific requirements.

1. Introduction

Service-Oriented Architecture (SOA) [5, 11] is a cutting-edge software engineering paradigm that provides a technical infrastructure for agile enterprises. Due to its interoperability and scalability potentials [5], SOA has gained increasing attentions in more than just business domain. Biomedical multimedia domain, among various disciplines advocating SOA, is an example that hopes SOA as their silver bullets for developing large-scale multimedia database software systems, featuring intensive nature of multimedia data and inherent complexity among diverse multimedia data across heterogeneous platforms.

However, besides heterogenous data and platform issues, such software systems may also have very specific and stringent yet sundry functional and Quality of Service (QoS) requirements. For examples, (i) HL71 standards, HIPAA2 policies, and FDA3 regulations are mandatory requirements that biomedical systems should guarantee; (ii) Biomedical multimedia data are usually of huge size and require QoS satisfactions provided by analysis, retrieval, storage and transmission services, to name a few; and (iii) Most importantly, such QoS satisfactions guaranteed by services tightly correlate to the contents of biomedical multimedia data. In order to tackle the biomedical multimedia systems’ abundant standards, policies and regulations, huge data size, and tight data/content-and-service correlation problems as well as to take the interoperability advantage to solve the heterogeneity issues and takes the scalability advantage for the purpose of future extensions, this paper presents a semantics- and data-driven SOA.

Our proposed infrastructure comprises five main aspects: (i) Services development for biomedical multimedia data. These services include, but not limit to, data analysis, transmission, and retrieval; (ii) Multimedia data annotations, comprising both automatic and manual annotations, will be supported; (iii) Ontology building, learning and reasoning. Biomedical multimedia data ontology will be built by extracting and organizing data annotations. Service and domain ontologies will be also described. Learning and reasoning may be later applied to decision making for finding appropriate services; (iv) Service discovery and selection. Appropriate services will be discovered and selected based on the semantics matching among data, service and domain ontologies; and (v) QoS optimization. For a biomedical multimedia system to accomplish data analysis, transmission, and retrieval tasks in compliance to specific use cases, orchestration languages [5] (e.g., WS-BPEL4) are needed to specify business process behavior based on services. However, many QoS are dynamically influenced by the data processed by services as well as the execution status and deployment environments of services at runtime. Introducing monitoring, learning, and adaptation mechanisms [11] may help tune up QoS or replace services dynamically. By achieving the five aspects, it is expected that the introduced semantics- and data-driven SOA will solve the aforemen-
tioned problems. Also, such an infrastructure could be utilized as a generic framework for constructing biomedical multimedia systems and as a domain-specific one with customized services developed and orchestrated.

The paper is organized as follows: QoS in SOA is summarized in the next section; Section 3 introduces the proposed infrastructure, followed by preliminary experimental results presented in Section 4; and finally, the concluding remarks and future directions are discussed in Section 5.

2. Quality of Service in SOA

Due to space consideration, this section only introduces major Quality of Service supports for SOA provided by the Windows Communication Foundation (WCF) platform [1]. For QoS support of Java EE 5, please refer to [14].

WCF is a distributed messaging platform under the .NET Framework 3.0 or later. It advocates SOA by supporting interoperability across different processes, machines, protocols, and non-WCF web services [1]. In order to provide satisfactory QoS, WCF introduces some advanced features to control throughput, security, and reliability. For throughput, PerCall or PerSession of instancing mode determines the lifetime a service allocated to requests. Concurrency mode and throttling behavior respectively determines if a service instance allows concurrent calls and control the load of each service by restricting the number of concurrent calls, sessions or service instances. For reliability, WCF introduces reliable session to overcome network failures and guarantee transmitting messages in order. Transactions are also present in WCF to coordinate a series of activities in a consistent state. Lastly, queued calls enhance reliability by guaranteeing that transmitting messages will be durably stored in queues until the message receiver is back online. As for security, when standard bindings that establish communication channels between clients and services are selected, default security configurations will be set. These configurations include six security modes (e.g., transport and message), four credential types (e.g., Windows Kerberos credentials), 16 encryption algorithms, and role-based authorization, to name a few [1]. All of the default settings can be customized based on service needs.

Although both WCF and Java EE 5 platforms provide QoS supports to services in various aspects, QoS supports to solve the problems of abundant requirements, huge data size, and data/content-service correlation in biomedical multimedia systems should be also considered. By taking data, services and domain requirements into consideration and semantically describe them in ontologies, it may be easier to discover suitable services and adaptively tune services to optimal in accordance to specific data.

3. Proposed Infrastructure

This section presents a semantics- and data-driven SOA for biomedical multimedia systems, specifically focusing on QoS issues that may be influenced statically and dynamically by both multimedia data and services correlatively. Two major use case scenarios are described below:

(i) End users perform both manual biomedical multimedia annotations and light-weighted video segmentation analysis at the end users’ site. All the data will then be updated to the server’s site.

(ii) Supposed an end user would like to request content-based multimedia data retrieval by using the updated data from (i) or inputting new annotations for retrieving specific data, an orchestration language will be used to specify the business process behavior in the following order: (a) Annotation services will be needed for processing updated or new input; (b) Services for ontology building, learning and reasoning will be performed; (c) Semantics matching services will use quantitative and non-quantitative approaches to discover the most suitable analysis, transmission and retrieval service(s), given that these services have been deployed and registered; (d) the orchestration language also defines the execution order of the selected services; and (e) the runtime environment of the orchestration language will bind and invoke the services and perform QoS optimization mechanisms. Each step in the use case scenarios will be explained in details in the following subsections.

3.1. Services for Multimedia Data

Because this paper only concentrates on the SOA for biomedical multimedia systems, services for such systems can be classified into analysis, transmission, and retrieval:

Analysis

The goal of analysis services is to develop data mining algorithms to discovery important patterns from biomedical multimedia data. Interesting analysis algorithms include: (i) Object location identification; (ii) Video segmentation; and (iii) Video classification. For (i), tracking algorithms are expected to quickly identify the location of objects, predict the objects in the sequential frames, and handle the transformation and occlusion. Our infrastructure implements Kalman Filter algorithm to track the moving of an object class [3]. For (ii), because most multimedia data are of huge size, segmenting them into smaller and meaningful chunks may help improve throughput or other related QoS. In the proposed SOA, two types of segments will be investigated: video segments that contain similar objects; and video segments whose images may not contain the same objects but share similar context (i.e., “semantic segment”). A parsing paradigm and a video segmentation algorithm have been introduced in [3]. (iii) may provide better multimedia database management to support other services. Two of our current focuses are: medical video event classification using shared features [3] and audio-visual event classification via spatial-temporal-audio words [3]. The first service provides a promising classification strategy for multi-class
video events, while the second one introduces a new representation of a video sequence (i.e., spatial-temporal-audio words) for classification.

Note that because video segmentation and video classification algorithms may also generate metadata as annotations for multimedia data, these implementations can be regarded as annotation services described in Section 3.2.

Transmission
The objective of transmission services is to offer efficient, reliable and secure transmissions in compliance with SOA to guarantee high quality, low latency, and security of data delivery. As described before, WCF provides a variety of transmission supports. However, such supports have to be configured by service developers at design time. For example, if an end user would like to submit a confidential skull image file to a remote site, he/she has no permissions to establish or re-configure the most appropriate transmission channel for the confidential data. In this infrastructure, transmission services are introduced – they can be regarded as autonomous transmission providers to fulfill data’s requirements. For example, the six security modes (i.e., None, Transport, Security, Both, Transport with Message Credential, and Transport Credential Only) along with the reliable mode (i.e., Reliable Sessions) and different encryption algorithms are wrapped as transmission services. When specific transmission requirements are requested, candidate services may be selected at orchestration time or at runtime.

Besides the aforementioned transmission services, our infrastructure is focusing on developing services in the following three categories:

(i) Efficient streaming: (a) Unicasting: how to stream large-size multimedia data while ensuring high quality and low delay; (b) Multi-path streaming: how to decompose multimedia data to multiple partitions of data streams and transmit each stream along disjoint paths. Three “Concurrent Transmission” services have been implemented to concurrently deliver segmented videos; and (c) Multicasting: how to transmit the same multimedia data to multiple users concurrently while maintaining sufficient bandwidth.

(ii) Secure streaming: Various security measure such as watermarking and its integration with efficient streaming protocols will be investigated.

(iii) Interactive streaming: When users make certain operations on the received video (e.g., scaling, rotation, forward, and backward), it is important that the server is aware of them and adapts its streaming session to accommodate, thereby significantly reducing the waiting time.

Retrieval
The objective of retrieval services is to efficiently assess large-scale biomedical multimedia databases using content-based image/video retrieve algorithms [4]. However, due to a wide range of transforming, smoothing, and rendering, the same object in different videos may have totally different size, shape, and textures. How to design algorithms that are invariant to the object scale, illumination change, texture change, and transformation is a challenging problem. Our current focus is to investigate algorithms that can handle shape invariance, introduce new similarity measurements, and extend the measurements to handle other variances (e.g., texture and illumination changes). The future plan is to introduce adapters to embrace existing work surveyed in [4] into our infrastructure.

Again, although there are only three types of services investigated, the loosely coupled design of autonomous services, interoperable message communications, and commonly agreed standards utilized in this infrastructure allow new services to be added and orchestrated easily in the future. All existing and new services should be described properly by WSDL [5], including their input and output formats, functionality provided, standards/policies/regulations followed, and QoS properties, to name a few. Finally, analysis and retrieval services are usually considered as a joint component in the content-based image retrieval community. Our infrastructure decomposes them because of the definition of “service” [5]. Such services can be easily composed together and considered as a single aggregator/composite service under our infrastructure following SOA principles.

3.2. Multimedia Data Annotation

In order to better annotate biomedical multimedia data, our infrastructure classifies biomedical multimedia annotations into four categories: data properties, contents, QoS annotations, and regulations.

Data Property Annotations
Data properties describe physical attributes of the data instead of the contents of the data. They can be further classified into text, audio, video and image properties. The followings are some example attributes used:

(i) Text Properties: file name, file size, file format, created date, last modified date, font, and font size.

(ii) Audio Properties: file name, file size, file format (e.g., mp3, avi), created date, last modified date, duration/length, bit rate, sample rate (e.g., 44Hz), channels, and layer.

(iii) Video Properties: file name, file size, file format (e.g., MPEG-2, flv), created date, last modified date, duration/length, resolution, number of frames, number of streams, average bit/second dedicated to a video stream, width, height, and supported players.

(iv) Image Properties: file name, file size, file format (e.g., bmp, jpeg), created date, last modified date, width, height, pixel format, and resolution.

Content Annotations
Content annotations using image/video content analysis have been an important topic for years. Annotations are obtained by either performing analysis services (e.g., video
segmentation and classification) to automatically generate useful analysis results or manually updating useful content information. Important automatic and manual annotation approaches for images and videos can be respectively found at [18] and [16]. Our current status is introducing a MPEG-7 based manual annotation portal adapted from the Caliph & Emir project [9]. This portal provides an interface to annotate data, contents, QoS, and regulations. Investigating new automatic annotation services or introducing adaptors for existing work in [16, 18] to fit into our SOA infrastructure is our future work.

**QoS Annotations**

In order to solve the data/content-service correlation problem, QoS requirements for data and contents should be also annotated. For example, if a skull image is expected to be transmitted to a remote site in a secure way, QoS annotations (e.g., minimum requirements of encryption algorithms) for such an image should be described. With QoS annotations, discovering and selecting appropriate services configured by WCF or Java EE 5 platforms may be easier. Figure 1 shows the screenshot for users to manually input QoS and Regulation annotations (described next). For security options, users can determine the security level of data from six security modes provided by WCF. Similarly, reliability, latency, and segmentation requirements can be also decided based on data’s needs. If users provide insufficient or incorrect QoS requirements information, reasoning about most appropriate QoS by using other annotations available in the data ontology is desirable.

**Figure 1. The screenshot of manual annotation.**

**Regulation Annotations**

The message development framework introduced at the HL7 website (http://www.hl7.org) follows HL7 standards to guarantee high quality messages exchange in healthcare environments. Such a framework is implemented using the object-oriented paradigm. Because most of the users of biomedical multimedia systems have biomedical background and may know specific standards to follow, manually annotating biomedical regulations for multimedia data is also introduced in our SOA: If data to be processed require to follow specific and stringent HL7 standards, HIPAA policies, and FDA regulations, the services to be selected for data process also need to follow them.

Annotating data, contents, QoS and regulations is our first step for orchestrating biomedical multimedia systems out of appropriate services. Such a step can be treated as data requirements elicitation either by automatic analysis services or by manual end user input. It can be also regarded as an extraction step “that supports acquisition of domain ontology from textual sources [12].” Followed by annotations, ontology building is the next step.

### 3.3. Ontologies

“Successful employment of semantic Web services depend on the availability of high quality domain and service ontologies [12].” Building high quality domain ontology requires following features: *generic* enough to be used in many service descriptions; and *rich* enough to describe the complex relationships existing in a specific domain [12]. Some existing ontology learning frameworks and tools have been introduced to establish domain ontology from textual sources (e.g., TextToOnto [8]), and others may use visual editors/frameworks to construct ontologies (e.g., Protégé [13]). For example, domain ontology for HL7 has been created using Protégé [13]. Protégé also provides reasoning API to help infer logical consequences from ontologies. A number of other ontology learning techniques (e.g., statistical-based, rule-based, and hybrid) have been comprehensively surveyed in [21].

For building service ontology, OWL-S [17] has been widely applied to describe service semantics, which facilitate service discovery, composition and invocation. Because of the correspondence between OWL-S and WSDL [17], XSTL[^6] can be used to transform from OWL-S to WSDL or vice versa. Note that because our infrastructure specifically concentrates on using data ontology (introduced later) to discover and select most suitable services based on service and domain ontologies, the expandable *serviceParameter* and *serviceCategory* profile attributes of OWL-S should offer sufficient information (e.g., the HL7 standards supported by and Quality Rate provided by a service), so that semantics matching among the three ontologies for the discovery and selection purpose can be achieved.

Besides service and domain ontologies, this paper introduces data ontology extracted from multimedia data annotations described in Section 3.2. Because multimedia data annotations are mainly described by textual sources (e.g.,

[^6]: XML Transformations: http://www.w3.org/TR/xslt.html
As mentioned before, one of the objectives of OWL-S is to facilitate service discovery. With expandable serviceParameter [17], more informative functional and QoS properties and constraints that a service can provide or is limited to can be described. For example, an analysis service may describe maximum file size and file formats it can process. Also, this service may mention the specific kinds of contents it can track/segment/classify and if the contents should be audio-enabled or not. Lastly, a transmission service may describe specific encryption algorithms it provides. Conversely, data ontology comprises the elicited “QoS requirements” as well as data and content properties of biomedical multimedia data either from users or analysis services. Such information can be used to match the semantics of a service described in serviceParameter.

For quantitative matching (e.g., latency), mathematical formulae computed along with the directions of a flowchart under given constraints are the most popular approaches (e.g., [19] and [20]). For non-quantifiable matching (e.g., security) among data, service and domain ontologies, semantics matching with ontology learning and reasoning described before may be more suitable. For example, suppose a biomedical multimedia video that describes a fruit fly’s flying motion requires a reliable HL7 message transmission, the semantic reasoning engine may infer to the most appropriate transmission and analysis services.

Some technical issues worth mentioning are: (i) Inference rules for semantics matching are very domain-specific. Namely, each biomedical multimedia domain (e.g., digital forensics vs. cell biology) requires its own inference rules and mathematical formulae to select the most appropriate service(s); (ii) There might be more than one suitable service to be selected. Services selected based on quantitative approaches may be ranked based on the computation results. For those services selected based on semantics matching, ranking of those services should be determined with the help of domain experts. Even worse, such ranking results may be applied case by case instead of just based on a specific domain; (iii) For both quantitative and non-quantitative matching, various functional and QoS properties may be weighed differently based on the importance to specific domains. How to express such importance into specific inference rules is a challenging task; and (iv) A service usually has more than one associated serviceParameter and these parameters may also have orthogonal or non-orthogonal associations [7] between each other. How to introduce and manage such tangled relationships is difficult.

3.5. QoS Optimization

In order for a biomedical multimedia system to accomplish data analysis, transmission, and retrieval in compliance to specific use cases, orchestration languages [5] are needed to specify business process behavior based on services. However, there might be more than one suitable candidate service. Also, QoS offered by a service are influenced by multimedia data, internal resource status and external deployment environments [2]. Lastly, business/system/application requirements are ever-changing and candidate services may not be available.

All of the above factors suggest that either the runtime environment of orchestration languages (e.g., [10]) or business processes (e.g., [6]) should be extended to support QoS monitoring, learning and adaptation as well as dynamic service replacement [10]. Our past work [2] introduced dynamic service adaptation by intercepting the Just In Time compilation events under the .NET platform. This is more a business process-related extension. Currently, our focus is to expend the runtime environment to introduce suitable mechanisms specific to the biomedical multimedia domain.

4. Preliminary Experimental Results

This section presents some preliminary experimental results for transmitting three multimedia files. Six transmission services are experimented with: (i) HTTP-based concurrent buffered transmission; (ii) TCP-based concurrent buffered transmission; (iii) HTTP-based sequential buffered transmission provided by WCF; (iv) TCP-based WCF-ChunCKing implemented by [15]; (v) TCP-based concurrent streaming; and (vi) TCP-based sequential streaming provided by WCF. All of the services are not in secure mode, and all TCP-based are reliable. The experiments are performed under four traffic conditions for downloading files: (a) Local Machine; (b) Very good – 23,733 Kb/sec; (c) Average – 4,404 Kb/sec; and (d) Poor – 1,955 Kb/sec. Table 1 summarizes the average experimental results of HTTP-based transmission for 60MB and 132MB files. Each file is evenly segmented into six pieces. The results show that our HTTP-based Concurrent Transmission service performs better small file transmission. Table 2 summarizes the average experimental results of TCP-based transmission for 60MB and 1.07GB files. The results show that WCF-ChunCKing service is the best for both 60MB and 1.07GB files. However, note that such a result highly depends on setting of the number of chunks (or chunk size) – the more the chunk number, the slower transmission. Due to buffer size limitation in WCF, (v) cannot be experimented here.

The experimental results shown in Tables 1 and 2 can
be specified in service ontology using serviceParameter of OWL-S, which can be used for semantics matching.

5. Conclusion

Introducing semantics into SOA is not a new idea. However, when developing biomedical medical systems under SOA, multimedia data semantics should be considered. With such semantics, most appropriate services for processing these specific data may be discovered, performed, tuned up or replaced as needed. Besides introducing data semantics, serviceParameter of OWL-S is expanded to fit into our needs of specifying services in more details with respect to the data semantics. Such service semantics provide support to match data semantics. Additionally, WCF-provided and customized transmission configurations are wrapped as services that can dynamically invoked. This mechanism may help optimize QoS on-the-fly. Lastly, our infrastructure may be regarded as a generic framework for constructing biomedical multimedia systems and as a domain-specific one with customized services developed and orchestrated.

Currently, all the five aspects are being continuously improved with new features/algorithms. Several existing frameworks, tools, and techniques are also investigated. Our hope is to introduce adapters so that these existing artifacts can be reused and deployed in our SOA.

References


---

### Table 1. HTTP-based

<table>
<thead>
<tr>
<th>File Size + Service + Traffic Cond.</th>
<th>Avg. Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>60MB + (i) + (b)</td>
<td>8.3735 sec</td>
</tr>
<tr>
<td>60MB + (ii) + (b)</td>
<td>8.8073 sec</td>
</tr>
<tr>
<td>132MB + (i) + (b)</td>
<td>18.7641 sec</td>
</tr>
<tr>
<td>132MB + (ii) + (b)</td>
<td>20.9302 sec</td>
</tr>
<tr>
<td>60MB + (i) + (c)</td>
<td>102.6517 sec</td>
</tr>
<tr>
<td>60MB + (ii) + (c)</td>
<td>173.7121 sec</td>
</tr>
<tr>
<td>60MB + (i) + (d)</td>
<td>348.0260 sec</td>
</tr>
<tr>
<td>60MB + (ii) + (d)</td>
<td>372.2708 sec</td>
</tr>
</tbody>
</table>

### Table 2. TCP-based

<table>
<thead>
<tr>
<th>Size + Service + Traffic</th>
<th>Chunk No.</th>
<th>Avg. Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>60MB + (i) + (b)</td>
<td>6</td>
<td>7.4954 sec</td>
</tr>
<tr>
<td>60MB + (ii) + (b)</td>
<td>6</td>
<td>6.0878 sec</td>
</tr>
<tr>
<td>60MB + (i) + (c)</td>
<td>1</td>
<td>3.3333 sec</td>
</tr>
<tr>
<td>60MB + (ii) + (c)</td>
<td>94</td>
<td>6.1804 sec</td>
</tr>
<tr>
<td>1.07GB + (i) + (a)</td>
<td>12</td>
<td>54.4281 sec</td>
</tr>
<tr>
<td>1.07GB + (ii) + (a)</td>
<td>12</td>
<td>13.6709 sec</td>
</tr>
<tr>
<td>1.07GB + (iii) + (a)</td>
<td>12</td>
<td>51.1992 sec</td>
</tr>
<tr>
<td>1.07GB + (iv) + (a)</td>
<td>94</td>
<td>26.2344 sec</td>
</tr>
</tbody>
</table>