An SOA Approach to Multimedia Applications

P S Subramanian
Tata Consultancy Services
Plot no.96, EPIP Industrial Area,
Whitefield road, Bangalore.
+91-80-6660 8460
subramanian.sitaraman@tcs.com

Hrishikesh Sharma
Tata Consultancy Services
Plot no.96, EPIP Industrial Area,
Whitefield road, Bangalore.
+91-80-6660 8603
hrishikesh.sharma@tcs.com

Prakasha T S
Tata Consultancy Services
Plot no.96, EPIP Industrial Area,
Whitefield road, Bangalore.
+91-80-6660 8567
prakasha.ts@tcs.com

ABSTRACT

The Multimedia Research group in TCS, Bangalore is currently formulating a Multimedia Application Framework (MAF) on the lines of Services oriented Architecture (SoA). The critical tasks involved in the application of SoA are *service identification* and *orchestration*. Both these tasks require a clear conceptual view of the application domain.

A review of basic concepts of SoA followed by a novel conceptual view of the Multimedia domain is presented. Recasting this view in the framework of SoA is then shown to provide the required service identifications and orchestration specifications.

An illustrative example expressed in BPEL4WS is also given.

1. INTRODUCTION

As the multimedia information processing techniques continue to improve, demand for such content in various forms and combinations continues to grow. The deep penetration of internet in our daily lives has further fuelled the demand for richer ‘multimedia experience’ delivery for people. The ‘multimedia experience’ can be achieved by combining information represented in multiple sensory modalities in plethora of ways, giving rise to a rich variety of multimedia applications, some of which cannot even be foreseen at this moment.

What should then be the strategy for delivering multimedia applications whose specifications can only known “just in time”?

The strategy should consist of two parts. In the first part, a conceptual study of the multimedia domain should be conducted to identify “atomic” capabilities out of which any application can be built up. Further, the second part should provide a composition framework which chooses a suitable set of capabilities and assembles them into a system to meet the “just in time” given specification. One could say that such a strategy supports “application agility”.

On the other hand, the goals of the emerging paradigm of Service oriented computing and the associated concept of Service oriented architecture are “the assembly of application components into a network of services which can be loosely coupled to create dynamic business processes and agile applications” [3].

It is the intent of this paper to exploit the similarity of these goals with the requirements of the strategy for implementing “just in time” specifications discussed above and evolve an SoA approach to Multimedia.

The paper is organized as follows: The next section briefly reviews the relevant concepts from SoA. The third section provides a novel conceptual analysis of the Multimedia domain and derives the definition of atomic capabilities for implementing a comprehensive set of multimedia applications. It also lays the foundations for composition of these capabilities through the concept of multimedia map. An example application and it’s representation through the multimedia map is also given there.

The fourth section recasts the domain analysis of the previous section into the lexicon of SoA and provides UML models of various conceptual entities arising in this process. The fifth section, for the sake of concreteness, expresses the example of section 3 in the business process execution language BPEL4WS.

The concluding section summarizes the contribution of this paper and hints at future work.

2. SERVICE-ORIENTED ARCHITECTURE

2.1 About SoA

Service-oriented Architecture is a concept that is at a higher level of abstraction than both object and component oriented architectures(Figure 2-1)[5].

TACTiCS – TCS Technical Architects’ Conference’06

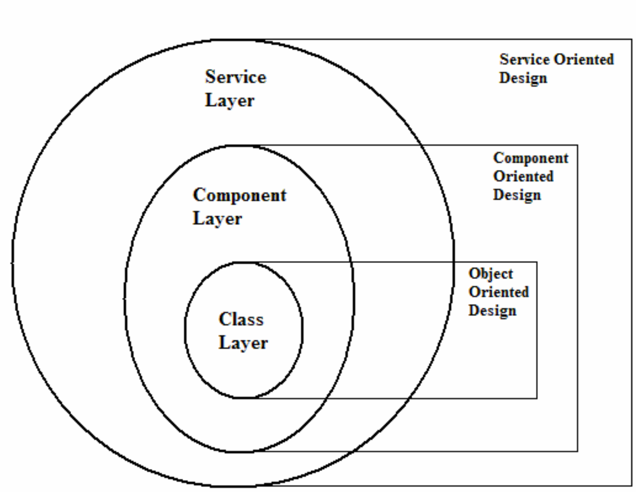

Figure 2-1 Layers of Design

A service is defined as a piece of business logic or functionality accessible via some interface. It provides computational or informational resources on request. Also, it has to be reusable so that multiple applications can use it for individual purposes as well as for purposes of interoperating with each other. Reusability imposes the requirement that its interface be designed such that it is composable.

In the case of remote delivery of a service, the delivery vehicle can be the world-wide web. Then one has to take into account protocol selection and syntax notations for service delivery. If the service environment is open, then one also needs to take care of protocol interoperation and syntax conversion.

2.2 Hierarchy of Functionalities

There are three levels of functional abstraction within the SoA paradigm

- *Operations*: These are transactions that represent single logical units of work. Semantically, they are directly comparable to object-oriented methods.
- *Services*: These represent logical groupings of operations which make sense at the level of the application domain. More precisely, assume that the concepts of the domain are expressed in terms of a domain ontology. Only those functionalities qualify as services which are expressible using the concepts of the domain ontology.
- *Business Processes*: These are a long running set of actions or activities performed with *specific business goals* in mind. Business processes typically encompass multiple service invocations.

2.3 Business Process Modeling

As mentioned earlier, services are composed with one another to satisfy a business goal. A way of defining applications is to define a set of '*business processes'*, each of which has a specific business goal. A service-oriented architecture is essentially a collection of communicating services [3]; the communication being required for composition and coordination. They can be of four kinds: orchestration, choreography, collaboration and workflow (see [1] for details). A business process is composed of a series of operations which are executed in an ordered sequence according to a set of business rules which are global constraints on the business processes. The sequencing, selection, and execution of services subject to the business rules is termed as service orchestration.

All existing BPM approaches can be leveraged as a starting point for SOAD [5].

2.4 Challenges in SoA

Anne-Marie Sassen et. al.[3] discuss in detail the certain challenges arising in the design of SoA. We will discuss some relevant aspects below.

1. *How should one develop services?*

Apart from default software development issues, the added dimension for service engineering is:

- how a service is *described* to users and application developers that want to use the service as it is, or want to integrate that service with others to develop a new service (*service description)*
- how these users *can find* what they require (*service discovery and service publication*)
- how can users *verify* that what is described is what is delivered *(service monitoring)*
- If users want to *compose* a new service on the basis of existing services, how can they do that *(service composition)*, especially on the fly, according to the needs of a user (*dynamic composition*)?

2. *How can we electronically represent all of the things that are relevant to a real world service?*

- We need a rich description for expectations, requirements, contracts, existing industry processes, etc. We also need ontological representations of the world context in which the services are delivered.

2.5 About BPEL4WS

BPEL4WS is a service description language under '*process collaboration*' category that is mainly based on XML for data description and WSDL for service description. The suffix of WS suggests that it relies quite a lot on web services based interfacing of activities to put up a business process. We don't need such interfacing in this paper; we choose this language only to provide an example.

The language has limited powers for describing computation done on the service nodes. However, it is quite powerful when it comes to describe synchronization of services. It uses rendezvous based synchronization construct in multiple forms for concurrent activities. The language currently models only binary (communication) association between services, though both blocking and non-blocking communication is allowed. The service binding is achieved by the notion of ports. One can write both abstract (model) and executable (model) of a business process. Here we are interested in abstract model only.

BPEL4WS does not cover issues related to service discovery, and hence its expressivity for service composition is also limited. One must use it along with UDDI to model a full-fledged service composition.

3. MULTIMEDIA INFORMATION PROCESSING

3.1 Multimedia – a conceptual analysis

3.1.1 Info – Actions

The evolutionary advantage enjoyed by (individual) humans over other organisms is primarily due to their ability to handle **information** - *generation*, *storage*, *processing*, *representation*, and *communication*. This applies to *human societies* too: societies that possess efficient processes for these tasks have a distinct advantage over those that do not. The importance of Information Technology (IT) is a consequence of this fact. We define the term info-action to denote **info**rmation handling **actions** of the type listed above.

The major *quantifiable* resources required to support the IT infrastructure are *processing* power, *communication* bandwidth and *storage* capacity. Due to the (economic) importance of IT, highly profitable industries have evolved to provide these resources (Figure 3-1).

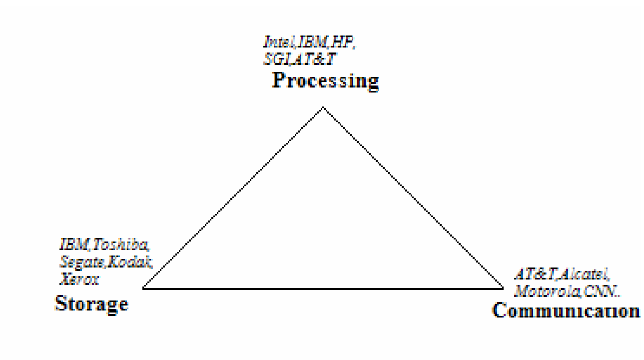

Figure 3-1 Some Info-actions and Companies Built Upon Them

There is an interesting phenomenon connecting these resources.

Though conceptually independent, they admit the existence of techniques, *which allow one resource to be traded against the others*.[2] This is quite important from an economic point of view, since *at a given point in time* the relative costs of the resources *might* show a large variance. Supposing that we are in need of a resource B which is costlier than A and C. In such a situation, it may be economical to use A and/or C in conjunction with techniques which trade off B for A and/or C. By annotating the edge AC with these techniques, one can derive the following additions to the diagram. If, for example, storage is costly, then we can use compression techniques. More generally, if the functionality at a vertex of a triangle is costly, then one can use the functionalities labeling the opposite edge for tradeoffs. Naturally, profitable industries have evolved around this aspect too(Figure 3-2).

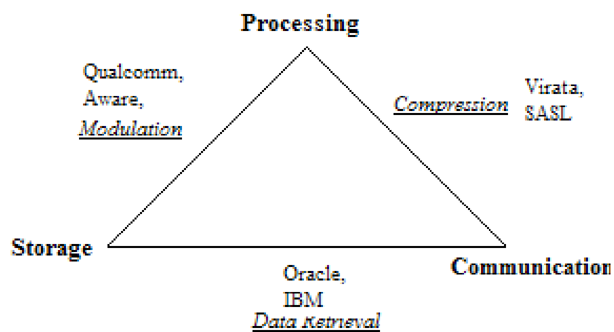

Figure 3-2 Companies Built on Info-action Trade-offs

In the context of societies- either of men or machines or a combination of both – the need to communicate information arises. Often, the sender can use the knowledge – a priori *or* learned during the course of communication - of the information processing capabilities of the receiver for optimizing the communication task. *In particular this may allow a reduction in the quantity of information to be transmitted*. This situation is quite generic. Hence "reduction" can be added to the set of *abstract* info-actions as shown in Figure 3-3.

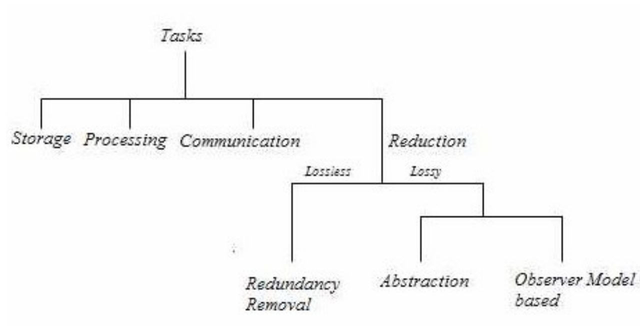

Figure 3-3 Info -Actions

3.1.2 Multi-Modality

In the discussion above, we used the concept of Information in a generic sense. However, Information can be *represented* at various degrees of *abstraction* and distinct *modalities*(Figure 3-4). Moreover, senders and receivers involved in communication of information could be subsets of humans and machines.

At the bottom of this Figure 3-4 one has the *concrete universal* representation corresponding to the bit level. In addition to the perceptual modalities corresponding to human senses, human information processing involves representation in Natural Language and Mathematics adding an additional cognitive dimension. This corresponds to *abstract universal* representation. One should think of the degree of abstraction increasing from the bottom of the diagram to the top.

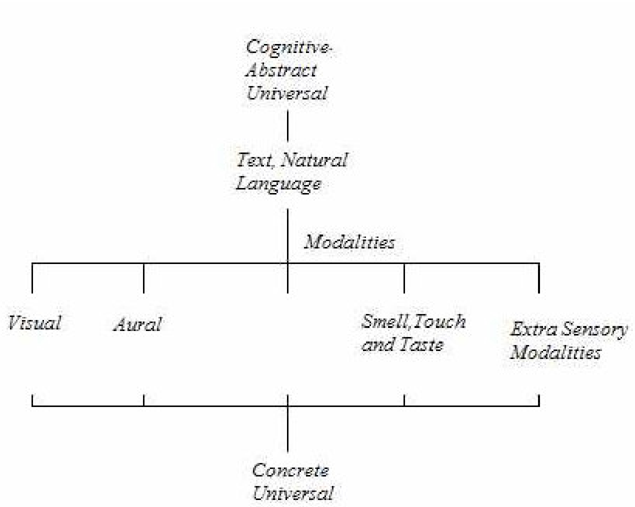

Figure 3-4 Hierarchy of Modalities

When humans communicate with each other they make use of all the modalities available to them. It would have been very natural if they could have used the same modalities to communicate with machines too. This was not possible until recently due to the economic and scientific considerations. The necessary hardware was too costly and/or our understanding of information processing in these modalities was too sketchy. However, progress in these "enabling technologies" has been considerable in recent years. Thus the field of Multimedia Communications has matured to the point of being *potentially* deployable in the real world. Certain gaps have to be filled before this potential becomes an actuality. It is with the filling of these gaps, the domain of multimedia is concerned with.

3.2 Aural Information Map

To illustrate the depth of the conceptual analysis carried out in the last section, we apply the insights obtained there to the aural domain(Figure 3-5). It should be pointed out that though this is just to a fragment of the domain of multimedia, non-trivial applications come to the surface. This is an indication of the power of this analysis methodology in "discovering" new business opportunities.

.

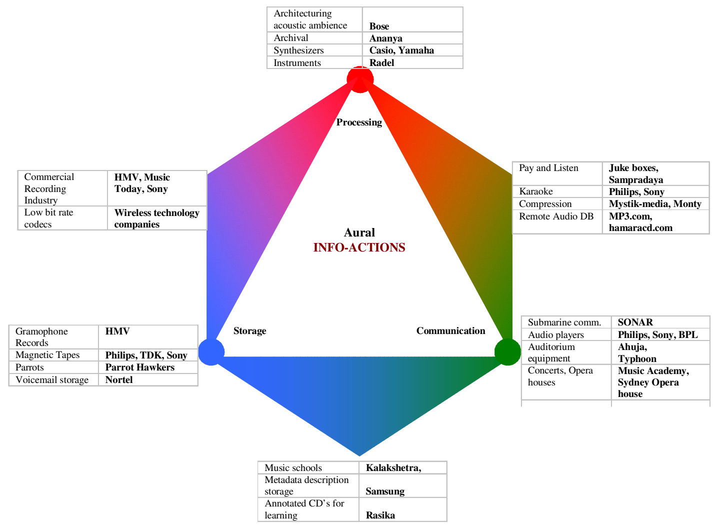

Figure 3-5 Aural Info-actions and tradeoffs

3.3 The Multimedia Map

Combining[1] the figures 3-1 and 3-4 we get a diagram which we label as the "multimedia map" (Figure 3-6).Due to the exhaustiveness of our analysis, this map contains all the required functionalities needed to implement any multimedia application at a suitable level of abstraction. Conversely, any specific application will involve tracing a set of paths through this map.

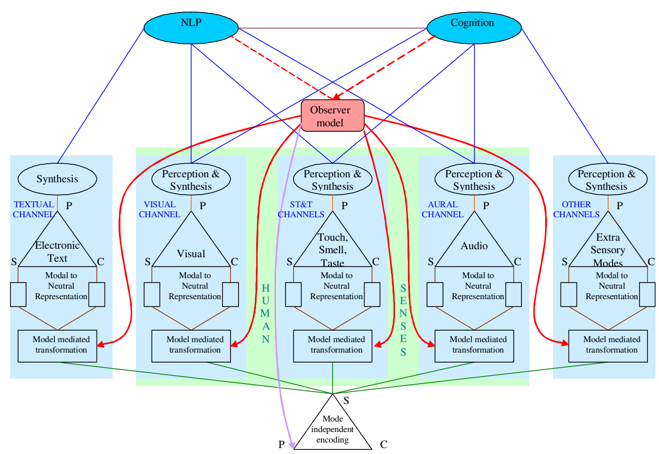

Figure 3-6 Multimedia Map

The model-mediated transformations in the figure correspond to the observer model based reduction shown in Figure 3-3. For drawing convenience, the info-actions processing, storage and communication are denoted by P, S and C respectively. Note that some arrows are unidirectional. The arrowheads show the

[1] Mathematically this involves taking Wreath Product of diagrams.

direction of control and/or information flow. All other lines are bi-directional.

This Multimedia Map has been used for business opportunities' identification and discovery of multimedia applications. More details are available in [4].

3.4 Example: Remote Lecture Delivery

Imagine a course on Multimedia being conducted at an University. To his consternation, the lecturer finds out that he is not only having a sore throat, but running fever also. It being an important lecture on info-actions, he decides to conduct the class sitting at home. So he picks up sheets of paper, scribbles the lecture details in sequence on them, and scans and transmits them as electronic text. In the classroom, a gizmo not only converts the received text into the *lecturer's* voice, but also simultaneously displays it on a big electronic screen. The control/data flow model involving various modalities for this application is shown in Figure 3-7.

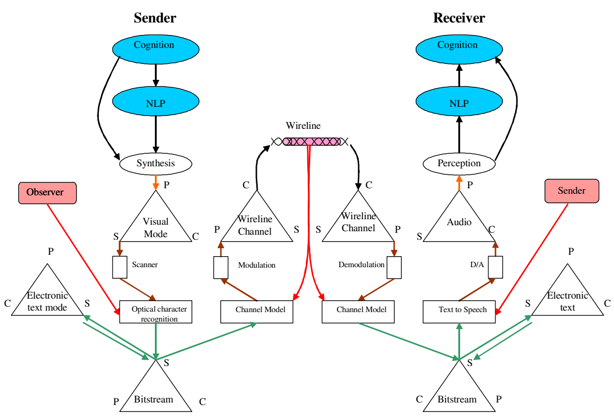

Figure 3-7 Remote lecture map

The sender info-actions refer to crude form of videotext transmission, while the receiver info-actions relate to audio and text management and reception. These actions could be integrated to synthesize a sender scenario, where cognition and NLP synthesizes professor's thoughts to the hand written text on paper. This paper is scanned to convert it to digital image format and fed to an Optical Character Recognition engine based on an observer model. The output of the OCR is electronic text, which is then sent over the wireline channel after FEC and modulation.

The receiver scenario mainly involves audio processing. The received signal from the channel is source- and channel-decoded, and then demodulated. This bitstream (eText) is fed to Text-to-Speech converter, then digital-to-analog converter, and played on the classroom speakers. The eText is further sent to the screen for displaying purposes using the display driver.

4. SOA APPROACH TO MULTIMEDIA APPLICATION FRAMEWORK

The multimedia application framework is being conceptualized in order to facilitate the application developers to utilize this tool and create/generate/develop/implement applications involving multimedia data. The aim is to provide a one stop solution to all multimedia-related projects at application development and deployment level.

As seen previously, the multimedia data could be in various single-mode forms such as Audio, Video, Touch, Smell, Taste, Textual or Extra-sensory information. It can also be in multi-modal forms [4]. This data can be processed, stored or communicated apart from creation and consumption. The area of focus for us as of now is *designing services* for multimedia *information processing*, *storage* and *communication*. Because of service-oriented architecture, requirements for *information representation* also creep in. In this section for sake of illustration, we cover efforts towards defining the conceptual requirements in the area of **Audio Processing Services.**

Table 1 provides (not an exhaustive list) a set of (audio processing specific) functionalities corresponding to the three levels of abstractions described in section 2.2.

This table is followed by a set of conceptual models described in UML notation for viewing the Audio processing services in the SoA framework. These models use the template prescriptions given in [3].

Operations	Services	Business Processes
Transforms • Z-transforms • Fourier • DCT • Hilbert • Goertzel • Wavelet **Time domain** • Up sampling • Down sampling • Correlation (Cross/Auto) **Measurements** • Amplitude • Power • THD • SNR • Phase distortion • Phase delay • Group delay **IIR Filtering** • Butterworth • Bessel • Elliptic • Chebyshev **FIR Filtering** • Parks-McClellan • Direct Calculations (raised cosine) • Windowing (Hanning, Hamming...) **Statistical Signal Processing** • Adaptive filtering (LMS) • Detection (Hypothesis) • Estimation (Bayesian, LMMSE, Minimum variance)	• Signal conditioning • Noise filtering • Multiple measurement s • Pitch detection • Speech/music identification • Silence detection • Melody detection • Rhythm detection • Spectral analysis • Linear Predictive Coding • Psycho acoustic modeling	• Speech / Audio compression/decompressi on • Speech synthesis • Music synthesis • Voice recognition • Speech enhancement • VOIP • Query by Humming

Table 1 SoA Functional hierarchy for Audio

Figure 4-1 onwards describe the composition of these info-actions using UML notation. The last, Figure 4-5, describe a *specialized* processing domain, the *audio analysis*.

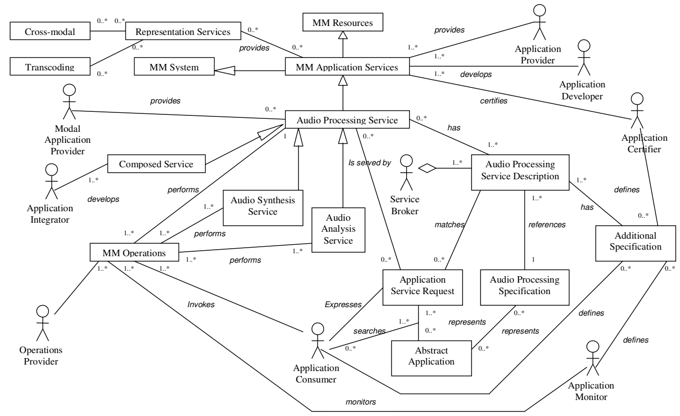

Figure 4-1 Conceptual Model of Audio Processing Services

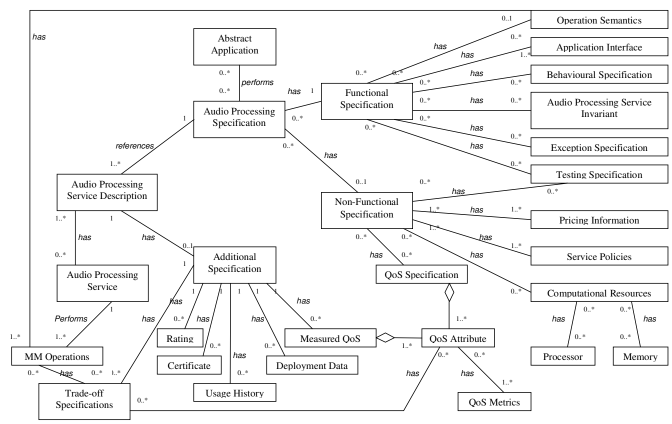

Figure 4-2 Conceptual Model of Audio Processing Service Description

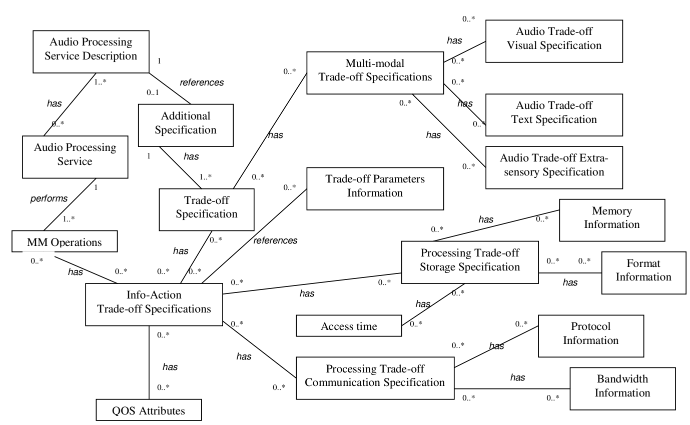

Figure 4-3 Conceptual Model for Audio Processing Trade-off Specification

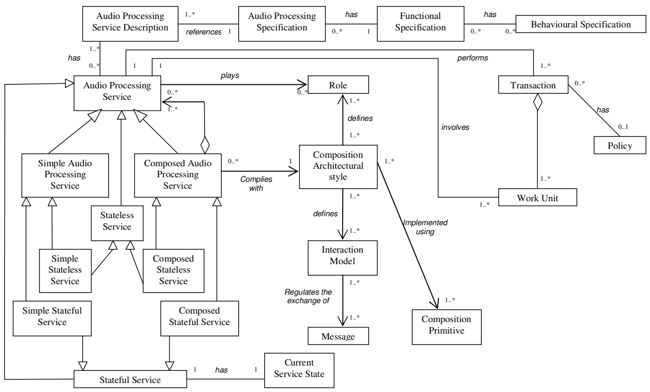

Figure 4-4 Conceptual Model for Audio Processing Service Composition

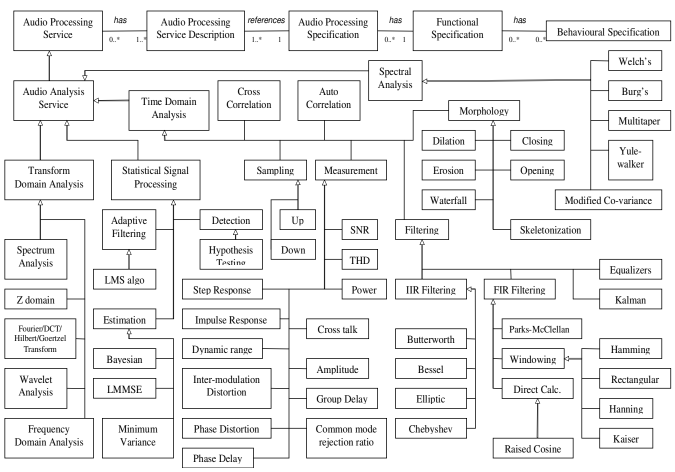

Figure 4-5 Conceptual Model for Audio Analysis Service Composition

5. MODELING EXAMPLE

We conclude our paper with modeling a portion of application example provided in section 3.4. As mentioned earlier, the services need not be distributed physically like the web-services model. It is true that the thrust on service-oriented architectures came from web-based distributed applications. However, our idea of designing an application framework is **not** to emulate *real-life services deployment*, but to provide a single platform on which rapid application development can happen. Also, we will only be modeling behaviour using *abstract processes* in BPEL4WS, where such details of distribution are not used in general.

5.1 Business Process Description

The (lone) business process involved in the example is smooth conduction of lecture. At a more concrete level, one can identify many services cooperating on either side(sender/receiver) to realize the business goal. On the receiver side, some of the prominent services involve signal post-processing(source/channel decoding), text-to-speech conversion, playing the speech and displaying the lecture material simultaneously. Assuming that text-to-speech conversion and playing it are intertwined, and that lecture material is received in series of frames, Figure 5-1 shows a simplified business process model of the receiver. Here, normal line denotes a sequential flow of activities, while dotted line implies a rendezvous of activities.

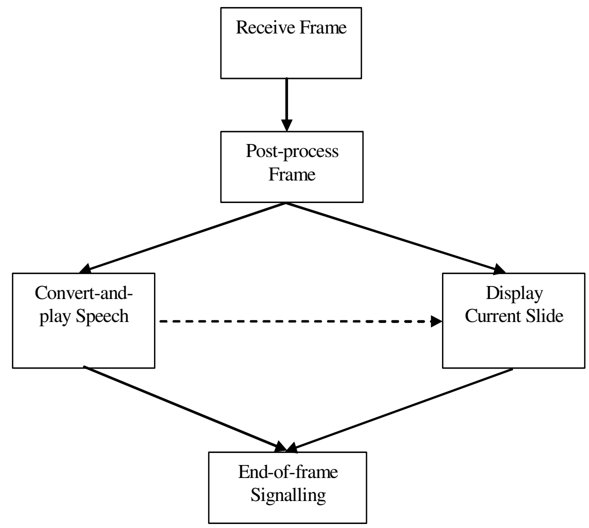

Figure 5-1 Simplified Process Model of Receiver

5.2 Formal Model using BPEL4WS

One first needs to put down the types used in modeling the business process as shown below.

```
<definitions
targetNameSpace="http://www.tcs.com/embedded/wsdl/multimedia"
xmlns:lectns="http://www.tcs.com/embedded/wsdl/multimedia"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns=”http://schemas.xmlsoap.org/wsdl”
      xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link" >

<message name="ETextFrameInMsg" >
   <part name="InputFrame" type="xsd:string" />
</message>
<message name="DecodedFrameMsg" >
   <part name="DecodedFrame" type="xsd:string" />
</message>
<message name="SpeechOverMsg" > </message>
<message name="DisplayOverMsg" > </message>
<message name="FrameOverMsg" > </message>
```



```
<portType name="processFramePT" >
  <operation name="processFrame" >
    <input message="lectns:ETextFrameInMsg" />
    <output message="lectns:FrameOverMsg" />
  </operation>
</portType>
<portType name="postProcessFramePT" >
  <operation name="decodeFrame" >
    <input message="lectns:ETextFrameInMsg" />
    <output message="lectns:DecodedFrameMsg" />
  </operation>
</portType>
<portType name="speakingPT" >
  <operation name="playLectAsSpeech" >
    <input message="lectns:DecodedFrameMsg" />
    <output message="lectns:SpeechOverMsg" />
  </operation>
</portType>
<portType name="displayingPT" >
  <operation name="displayLecture" >
    <input message="lectns:DecodedFrameMsg" />
    <output message="lectns:DisplayOverMsg"/>
  </operation>
</portType>

<plnk:partnerLinkType name="processingLT" >
  <plnk:role name="processingService" >
    <plnk:portType name="lectns:processFramePT" />
  </plnk:role>
</plnk:partnerLinkType>
<plnk:partnerLinkType name="postProcessingLT" >
  <plnk:role name="decodingService" >
    <plnk:portType name="lectns:postProcessFramePT" />
  </plnk:role>
</plnk:partnerLinkType>
<plnk:partnerLinkType name="speakingLT" >
  <plnk:role name="speakingService" >
    <plnk:portType name="lectns:speakingPT" />
  </plnk:role>
</plnk:partnerLinkType>
<plnk:partnerLinkType name="displayingLT" >
  <plnk:role name="displayService" >
    <plnk:portType name="lectns:displayingPT" />
  </plnk:role>
</plnk:partnerLinkType>

</definitions>
```

One can the put down the abstract model of business protocol that the services within the business process follow, while coordinating with one-another, as follows.

```
<process name="RemoteLectureProcess"
targetNameSpace="http://www.tcs.com/embedded/internal/lecture
Example" suppressJoinFailure="yes" abstractProcess="yes"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-
process"
xmlns:lectns="http://www.tcs.com/embedded/wsdl/multimedia" >

  <partnerLinks>
    <partnerLink name="processing"
patnerLinkType="processingLT" myRole="processingService" >
    </partnerLink>
    <partnerLink name="postProcessing"
          partnerLinkType="postProcessingLT"
  partnerRole="decodingService" >
    </partnerLink>
    <partnerLink name="speaking"
patnerLinkType="speakingLT" partnerRole="speakingService" >
    </partnerLink>
    <partnerLink name="displaying"
patnerLinkType="displayingLT" partnerRole="displayService" >
    </partnerLink>
  </partnerLinks>

  <variables>
    <variable name="InputFrame"
messageType="lectns:ETextFrameInMsg" />
    <variable name="DecodedFrame"
         messageType="lectns:DecodedFrameMsg" />
    <variable name="SpeechOverSignal"
         messageType="lectns:SpeechOverMsg" />
    <variable name="DisplayOverSignal"
            messageType="lectns:DisplayOverMsg" />
    <variable name="FrameOverSignal"
         messageType="lectns:FrameOverMsg" />
  </variables>

  <sequence>
    <receive
partnerLink="processing"portType="lectns:processFramePT"
```

```
            operation="processFrame" variable="InputFrame" >
      </receive>

         <invoke partnerLink="postProcessing"
portType="lectns:postProcessFramePT"
operation="DecodedFrame" inputVariable="InputFrame"
outputVariable="DecodedFrame" >
      </invoke>
      <flow>
         <links>
            <link name="speaker-to-board" />
         </links>
         <sequence>
            <invoke partnerLink="speaking"
                  portType="lectns:speakingPT"
                  operation="playLectAsSpeech"
                  inputVariable="DecodedFrame"
                  outputVariable="SpeechOverSignal" >
               <source linkName="speaker-to-board" />
            </invoke>
         </sequence>
         <sequence>
            <invoke partnerLink="displaying"
                  portType="lectns:displayingPT"
                  operation="displayLecture"
                  inputVariable="DecodedFrame"
                  outputVariable="DisplayOverSignal" >
               <target linkName="speaker-to-board" />
            </invoke>
         </sequence>
         </flow>
      <reply partnerLink="processing"
           portType="processFramePT"
         operation="processFrame" variable="FrameOverSignal" >
   </sequence>
</process>
```

6. CONCLUSIONS

In this paper, we have shown how service oriented thinking helps in structuring the development of a Multimedia Application Framework. While the usual context of application of SoA is the provision of agility to business software in the face of changes to business processes, the principles underlying SoA can also be put to use in quick generation of "proof of concept" prototypes for a variety of Multimedia applications.

Future work will involve the implementation of the identified services within the relevant standards as well as the design of a domain ontology based orchestration engine in the context of developing a Multimedia Application Platform.

Another interesting, and in our opinion a crucial, direction of future work is the development of systematic methodologies for co-ordinatisation of the (semi-structured) space[2] of atomic services for specific domains of application. Further, this co-ordinatisation could effectively be used in the synthesis and validation of orchestration and choreography.

In this paper, we gave a taste of this methodology in the context of Multimedia. This methodology has also been proposed to an European Commission funded consortium, of which TCS is a member, in the context of a project on Autonomic Telecommunications. We are currently in the process of developing a general mathematical[3] theory based on these examples. Since, TCS is involved in many different Verticals, we expect that research along this direction will be quite valuable for *systematic* introduction of SoA methodologies in these Verticals.

REFERENCES

[1] Munindar P. Singh and Michael N. Huhns. *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley and Sons, 2005.

[2] H.L.Resnikoff, R.O.Wells Jr:: Wavelet Analysis- Springer,1998.

[3] Anne-Marie Sassen, Charles Macmillan. *The Service Engineering Area: An Overview of its current state and a vision for its future.* European Commission (Information and Media Directorate-General, Network and Communication Technologies) Report, 2005.

[4] Venkatasubramanian V., Prakasha T.S., P.S. Subramanian, *Advanced Multimedia Initiative*. TCS Internal Document, 2003.

[5] Olaf Zimmermann, Pal Krogdahl, Clive Gee. *Elements of Service Oriented Architecture and Design*. IBM Whitepaper, 2004.

[6] World Wide Web Consortium. *Business Process Execution Language for Web Services.* Recommendation version 1.1, 2003.

[7] World Wide Web Consortium. *Namespaces in XML.* Recommendation version 1.0, 1999

---

[2] Co-ordinatisation of a semi-structured space will involve, in general, a set of "axes" indexed by a Pomset (Partially ordered Multiset).

[3] The Wreath product construction used in building the multimedia map is part of this theory.