Journal of Theoretical and Applied Information Technology

www.jatit.org

AN EVALUATION MODEL FOR THE CHAINS OF DISTRIBUTED MULTIMEDIA INDEXING TOOLS RESPECTING USER PREFERENCES

[1]Bassem HAIDAR , [2]Bilal CHEBARO, [3]Hassan WEHBI

[1]Asstt Prof., Department of Computer Sciences, Faculty Of Sciences, Lebanese University, Lebanon

[2]Assoc. Prof., Department of Computer Sciences, Faculty Of Sciences, Lebanese University, Lebanon

[3]Department of Computer Sciences, Faculty Of Sciences, Lebanese University, Lebanon

E-mail: bassem.haidar@ul.edu.lb, bchebaro@ul.edu.lb, Hassan.wehbe@live.com

ABSTRACT

In this paper we present an evaluation model for the chains of multimedia indexing tools that takes into consideration the user preferences in order to enable him to choose the most suitable chain of tools respecting his preferences.

Two approaches are proposed to extract indexing chains. The first one suggests building the whole graph for all indexes present in the platform. The second approach deal individually with each user request: only the corresponding sub-graph is built.

Keywords: *Automatic Multimedia Indexing, Evaluation model,*

1. INTRODUCTION

The large amount of multimedia documents accessible and the need to access its essential content precisely to satisfy user's preferences promote the development of techniques for multimedia indexing and searching. During the last decade significant progress has been made in multimedia indexing and retrieval.

Despite the progress and effectiveness of indexing tools developed, many indexes need to be produced by a sequence of indexing tools that can be probably distributed on multiple sites.

For example in order to extract English spoken text from a video, we need first to detect human presence in the video. Second, we use an English detection tool in order to detect English speech content, and finally we use specific words recognition in order to transcribe the speech into text. The automatic multimedia indexing framework, allows, thanks to the chaining algorithm, to find the sequence of tools able to produce an expected index.

So, many collaborative multimedia indexing projects have been developed [3], [4], [5], [6], [7], [8], Some of these projects were dedicated for multimedia indexing while others were designed for workflow composition purposes.

In a previous work [1] [2], we proposed a generic wrapper in order to integrate legacy indexing tools into the platform, and we focused on the dynamic chaining and composition problem of heterogeneous multimedia indexing tools, the goal was to find dynamically the workflow able to generate new multimedia indexes that cannot be generated by a simple indexing tool. The proposed platform didn't allow the user to choose the more appropriate result, i.e. the chains that answer these preferences more.

In this paper we present an evaluation model for the chains of indexing tools that takes into consideration the user preferences in order to enable him to choose the most suitable chain of tools respecting his requirements.

In order to evaluate a sequence of indexing tools, we must define parameters and criteria that characterize an indexing service. Many researches has focused on defining a QOS parameters to characterize a distributed service or component and later to characterize a workflow, the most famous proposal is that of Cardoso [9][10] or Zeng [11], that was adopted by many other works [12] [13] [14]. In our work we adopt the Cardoso proposal that consists of a successful QoS model including a comprehensive objective function and formal definitions of major quality attributes [15].

This paper is structured in four paragraphs:
The first one (global architecture) provides a quick description of our platform; we can found a presentation of its global architecture and description of its two main class of services: centralized one and distributed ones. The second paragraph (tools evaluation model) give a quick review of evaluation parameters used to evaluate chains representing composite service. The third paragraph (adaptive chains extraction) describes our algorithm proposed to permit an adaptive extraction of indexing chain in a distributed environment corresponding into the best fit of the user's requirement. The fourth one is the conclusion

2. GLOBAL ARCHITECTURE

The platform is composed of distributed services and a central server that implements two principal services: the repository service and the access service (see Figure 1)

2.1 Chaining Algorithm

We use the SOAP protocol [16]. It presents the advantage of posting data over the HTTP protocol. On the Web service side, the SOAP request is processed and the result is sent back to the client using a SOAP response.

We use Apache Axis [17] for development. It provides SOAP support for Apache Tomcat application servers. Apache Axis use SAX for XML parsing.

We use a DataHandler [18] object to transfer the multimedia content. With this DataHandler, the Java Activation Framework provides a serialization and deserialization functionality and attaches it automatically to the sent SOAP message.

2.2 Opened platform, tool wrapper

A multimedia indexing tool is in general an executable program characterized by a specification file (XML format) written by the owner or the developer of the tool. The specification file contains information needed to execute this tool like the number and the type of input data, the number and the type of the output data, options needed for the execution, etc.

We have developed a wrapper, which is a package of code able to read the specification file, and then to launch the tool with the required command line which looks like:

Prog_name option1 imput1 input2 output1 output2

The following example is a specification of an audio segmentation tool:

```
<Component ComponentRole="Index">
<Description>To localize segments of Speech, Music,
Noises and Silences in an audio file</Description>
<Authors>J.Pinquier</Authors>
<Version>v123.2b</Version>
<Input>
  <InputData>
   <input_type>type1</input_type>
   <FileFormat>wav</FileFormat>
   <StorageInputPath>/Input_Path</StorageInputPath>
  </InputData>
</Input>
<Output>
 <OutputData>
  <output_type> AudioSegmentContent</output_type>
  <FileFormat>xml</FileFormat>
  <StorageOutputPath>/output_Path</StorageOutputPath>
 </OutputData>
</Output>
<Options>
<Option>option1</Option>
</Options>
</Component>
```

Figure 2 specification file of an audio segmentation tool

This generic wrapper is independent from the type of the tool, and only the specification file is required.

2.2.1 Tool invocation

The generic wrapper allows the invocation of multimedia indexing tool. Using its specification file, the wrapper builds the command line required to launch the tool

2.3 PLATFORM SERVICES

The platform is composed by two main centralised services, and a multiple of distributed services

2.3.1 Centralised Services

The main services are:

A. *Repository service*

The repository service allows the registration of new services on the platform. A new tool will be integrated on the platform by sending towards the repository service its specification file. Semantic types are inspired from the ones defined by the

MPEG-7[19] standard when it is possible. Other semantic types have to be added to provide pieces of description which are not available in MPEG-7. We implement some XML research functions in order to browse the XML specifications.

B. *Access service*

The access service allows a centralised access to the platform; its interface provides all the functionality of the platform:

- We can search for a given service using different parameters like its output or its input data type, or its name.
- We can search for available multimedia documents.
- We can launch a service by selecting this service and its required input data.

2.3.2 Distributed Services

A. Multimedia indexing services

a. *Indexing service*

The goal of the platform is to allow the access of distributed multimedia indexing tools, an indexing service can interface one or several indexing tools, developed by a single research team or specialized in a single media analysis like audio segmentation, or speech transcription. The indexing service must allow a user or another indexing service to use one of the handled indexing tools. Each indexing service is identified by a service identifier by the server side, and contains a table of "indexing tool representation" objects. Services have a SOAP-RPC interface that implements the service functionality.

b. Multimedia documents service

A user can add a multimedia document access service to the platform. As a multimedia indexing service, the multimedia documents service allows to access to multimedia contents, and has a specification file that describes the type of contents. A SOAP interface is defined to transfer data from and to this service. We use DataHandler object to transfer multimedia documents.

3. TOOLS EVALUATION MODEL

Cardoso [2] define four main quality attributes to characterize a single web service, these attributes can be used to characterize a workflow or a composite service, for a task t:

Service Time (ST) can be defined as the total time between the time of receiving a request by a service and the generation of results. The service time can be decomposed into two factors, the delay time and the processing time.

The first is called the delayed time (DT) is the time needed for a task to be treated, i. e. It is the additional time required for a request to be processed by a tool.

Processing or Execution time (PT) is the essential time required to process a requested task.

Therefore, the response time of a task t can be calculated as follows:

$$ST(t) = DT(t) + PT(t)$$

Reliability (R) is the probability that the service is executed when the user asks for, and therefore it is calculated based on failure rate. The value of reliability is calculated by the following formula:

$$R(t) = 1 - \text{failure rate}.$$

Execution cost (Ec) refers to the amount of money that a service requester has to pay for executing an operation

Fidelity (F) It is a quality measure; it refers to the characteristic of a good product or good service. Fidelity is often difficult to define and measure because it's subjective judgments. It is often predicted, when possible. They are not a single parameter can describe the precision of a tool, a service can have a vector fidelity composed by a series of attributes, each attribute refers to a property or a feature of the service being created, processed, or analyzed. In the case of indexing multimedia Recall / precision can are used to evaluate some indexing tools, but we cannot consider that they are the only parameters used to evaluate a tool.

In order to calculate the values of the proposed parameters for a chain of indexing tools, reducing the chain to be considered as a single tool is often used [11]. The chain of tools is modeled as a DAG (Direct Acyclic Graph). This reduction consists in merging the serial and parallel tools following predefined methods of reduction (Figure 4). These methods will be defined for each parameter.

The service *time* of a single tool is the universal time between the start of execution and the beginning of result appearance. Otherwise, the time required to execute a sequence of tools in parallel, depends on the slowest tool, however this proposal is not valid for a serial sequence. In this case the total time needed is the sum of execution time of all chained tools. From which these formulas:

Serial: $T_{chain} = \sum_i (Execution\ time)_{Tool\ i}$

Parallel:

Vol. 13 No.2 March, 2010 pp (139 - 147)

T_{chain}=$Maximum\ \{(Execution\ time)_{Tool\ i}\}$

The execution *cost* is defined as the total amount of money asked by the owner's tool to execute it for the user. The same definition is for a chain. Since all the chained tools must be executed, so it does not matter how these tools are chained (serial or parallel), so the cost is the sum of cost of all tools:

$$Cost_{Chain} = \sum_i [\ (Nombre\ of\ use\ of\ Tool\ i) * Cost_{Tool\ i}\ ]$$

The probability of a tool is executed once user asks it is called ***reliability.*** This factor is very important because the failed execution of a tool, part of a chain, will cause the failure of the entire chain. Because of that this parameter is calculates as following:

$$Reliability_{Chain} = \prod_i (Nombre\ of\ use\ of\ Tool\ i) * Reliability_{Tool\ i}$$

Finally, the most quality measure is the *fidelity.* It is the rate of good service and accurate result provided by an indexing tool. So, it's hard to be calculated accurately; it's predicted most of the time. There is no single parameter able to describe the fidelity of a tool; it is a set of quality fields. In the indexing case, we can use the parameter of recall and *precision*, but it is not the single fields of quality. It is calculated like the reliability because all of them are a rate with a value between 0 and 1.

$$Fidelity_{Chain} = \prod_i (Nombre\ of\ use\ of\ Tool\ i) * Fidelity_{Tool\ i}$$

4. ADAPTIVE CHAINS EXRTACTION

5.1 Evaluation criteria

Evaluation of an indexing tool for a tool or a chain of tools using the parameters defined above is a static solution independent to the user preferences; it also presents a major difficulty when comparing two tools or two chain of tool due to the presence of four different parameters. Our idea is to assign weightings to each of the parameters in order to calculate a single value score, the total score, and also to respect the user's preferences by leaving to them the choice to allocate coefficients suitable for evaluation according to their preferences

Since each user has preferences to choose between a set of chains able to generate a specified index, we must enhance our work in order to help user making the best decision corresponding to its preferences. So, user must set a vector of coefficients for each tool ($Coef_{Time}$, $Coef_{Cost}$, $Coef_{Reliability}$, $Coef_{Fidelity}$), the sum of these coefficients must be one. For example, a user preferring a chain with a lowest price mainly and the best time secondly will choose these coefficients: (0.3, 0.5, 0.1, and 0.1).

The problem appearing now is the following: what is the significant of a high score? It indicates a good chain or a bad one? Defined rules and formulas for *Time* and *Cost* assign large values for a bad chain, otherwise, big values for *Reliability* and *Fidelity* indicate a worst chain, and vice versa.

In order to give the same signification for all parameters, we must modify our score formula. Since the *reliability* and the *fidelity* are complement of one, we can reverse their value by using the complement of one instead they real values. From which the final score formula:

$$Score_{Chain} = Coef_{Time} * Time_{Chain} + Coef_{Cost} * Cost_{Chain} + Coef_{Rel} * (1 - Reliability_{Chain}) + Coef_{Fait} * (1 - Fidelity_{Chain})$$

Now, a big score is corresponding to a bad chain and small one is for a good chain.

5.2 Chains extraction

Back to the main objective of our work, we plan to extract a set of chains able to generate a given index. Once user ask the chains able to generate an index ***x***, we must explore all available and *compatible* indexing tools to combine them, in a way that the generated index be ***x***. Two indexing tools are called compatible, if the output index of one of them is one of the input indexes of the other.

The main part of the algorithm is the identification of the chains able to generate the requested index: two approaches are applicable: Once again, user requests the chains able to generate an index ***y*** and then ***z***, should we re-explore all the indexing tools every time we need a chain? Here, two methodologies appear:

5.2.1 One graph for all indexes

The first consist of, the creation during the construction of the platform, a graph representing all the relations of compatibility between the tools available on the platform.

A. Advantages

- Save the time of chain search: Once the graph is built, chains extraction's problem is reduced to a simple browse of the graph, which can be a very simple and fast search algorithm.
- The constructed graph, represent all possible solution that can be generated by all available indexing tools available on the platform.
- No additional steps are needed for identification chains generating new indexes requested by the user

B. Disadvantages

- Working with a large graph, open the door to the updating problem. It is hard to add a new tool to the graph, and too hard to eliminate an existing one. Because we have to browse and reconnect all the tools affected by this modification.
- In spite of time economization during the chain extraction, a long time and not bad resources are needed in the construction time, which can be expensive.

The figure 5 is an example of a graph that represents all possible compatibility relation between all existing indexing tools on the platform

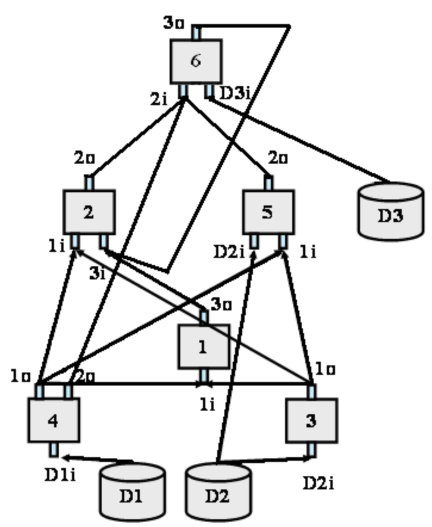

Figure 5

5.2.2 One graph per index

The second, it follows this following rule: to generate a given index a specific set of tools may have contributed to generate it. Our idea is to construct a graph that contains only the tools necessary to generate this index. This graph will consist of one or more chains each of which can produce the requested index. (Figure 6).

A. Advantages

- Graph built in this method is simple and dynamic in term of updating and browsing. (this graph can contain one or more chains with a minimum number of tools)
- As this graph offer the chain able to generate a single index, it is formed by the minimum and required number of tools. So, its time of construction is much reduced.
- In addition, if we save the chains extracted by each use of this method on many index, we can build a good database of chains able to generate diverse indexes.

B. Disadvantages

- The main inconvenient of using this method, is the large time needed every time we build a graph for a given index.

The figure 6 is a graph that represent all possible compatibility relations between existing indexing tools that allows the generation of a given index (type 3)

The indexing platform is an open platform; by consequence the integration of new tools into the platform will be frequent. Basing on the positive and negative side for the proposed methodologies, we will choose the second one which is more suitable to our target.

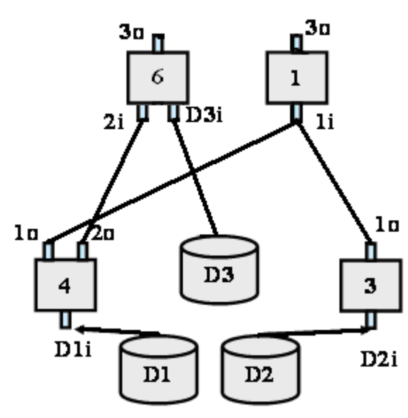

Figure 6

5.2.3 Score calculation

Vol. 13 No.2 March, 2010 pp (139 - 147)

After construction of the graph able to generate a given index, the next step is to extract all possible chains through a multiple traversals of the graph. The calculation of the score on each one of the chain result is carried out in application by applying the parameters reduction rules explained above in this paper, taking into account user preferences. Our chaining engine will automatically generate all the chains solutions sorted in ascending order of their score.

5.2.4 Execution example

We present an example of results produced by the implementation of the evaluation model, we define for example a set of resources indexing (indexing tools, multimedia primitive documents), and we try to extract the chains able generate a specific index type using the available tools.

The platform is composed of seven tools and two multimedia documents, these resources use four types of metadata (numbered from one to four) and two types of primitive multimedia data (D1 and D2). The types of input and output types of each tool are presented in Figure 7.

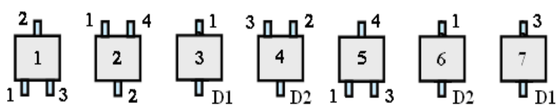

Figure 7

The evaluation parameters of each tool are shown in Figure 8:

```
Name : O1
Time = 1.6, Cost = 25.0, Reliability = 0.95, Fidelity = 0.9

Name : O2
Time = 1.5, Cost = 27.0, Reliability = 0.93, Fidelity = 0.88

Name : O3
Time = 1.0, Cost = 30.0, Reliability = 0.89, Fidelity = 0.91

Name : O4
Time = 6.0, Cost = 28.0, Reliability = 0.8, Fidelity = 0.95

Name : O5
Time = 5.6, Cost = 38.0, Reliability = 0.94, Fidelity = 0.9

Name : O6
Time = 5.5, Cost = 18.0, Reliability = 0.87, Fidelity = 0.92

Name : O7
Time = 5.5, Cost = 21.0, Reliability = 0.89, Fidelity = 0.94

Name : D1
Time = 0.0, Cost = 0.0, Reliability = 1.0, Fidelity = 1.0

Name : D2
Time = 0.0, Cost = 0.0, Reliability = 1.0, Fidelity = 1.0
```

Figure 8

In order to retrieve solutions chains able to generating the index 4, our algorithm will find first the relationship compatibility between tools (build the graph) starting from the index 4 (tools 2 and 5). The resulting graph is in presented Figure 9.

Scores are calculated based on the preferences of the user who specified the coefficients of the evaluation parameters as follows:

Time Coefficient = 0.3,
Cost Coefficient = 0.4,
Reliability Coefficient = 0.2,
Fidelity Coefficient = 0.1.

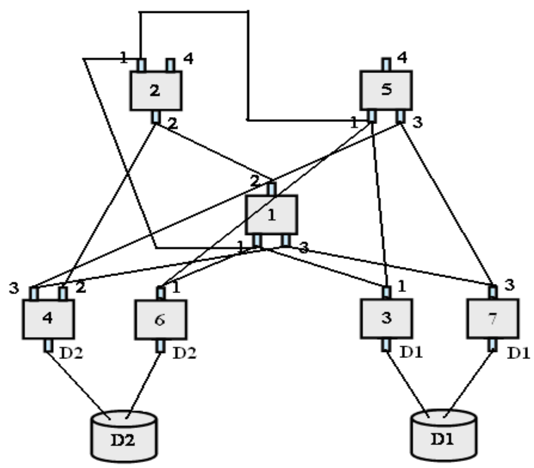

Figure 9

Once the graph is built, the algorithm tries to extract the set of solutions from the graph. The chains found by the algorithm are described in Figure 10

```
solutions :
1. O2->[O4->[D2]]
        Score => (64.51924340467895) / 100

2. O5->[O3->[D1],O7->[D1]]
        Score => (48.229701702339476) / 100

3. O2->[O1->[O3->[D1],O7->[D1]]]
        Score => (48.098203372324534) / 100

4. O5->[O6->[D2],O4->[D2]]
        Score => (46.78952573419612) / 100

5. O2->[O1->[O6->[D2],O4->[D2]]]
        Score => (46.74653240418118) / 100

6. O5->[O2->[O4->[D2]],O4->[D2]]
        Score => (32.09067100049775) / 100

7. O5->[O2->[O1->[O3->[D1],O7->[D1]]],O7->[D1]]
        Score => (18.949060865514085) / 100

8. O5->[O2->[O1->[O6->[D2],O4->[D2]]],O4->[D2]]
        Score => (15.166715039999985) / 100
```

Figure 10

5. CONCLUSION

In this paper we discuss the problem of services composition in distributed environment for multimedia indexing and retrieval purposes. We show the limitation of the classical approaches taking in account only the compatibility criteria between successive indexing tools and ignoring to rank different possible chains.

So we suggest using evaluation criteria that already used for evaluation of composite services. We proposed an algorithm exploring the graph of tools composition and evaluating according to a set of user defined criteria. Two approaches for graph exploitation were analyzed. The second approach was implemented and tested.

Despite the major disadvantage of the first approach, we believe that its implementation and usage according to some update protocol can provide excellent results: Collect periodically information about new tools added to the platform, update the graph. The graph could be proposed as an additive service provided to the platforms to the users

REFRENCES:

[1] B. Haidar. Services d'Indexation Multimédia Distribués. PhD thesis, Univ. Paul Sabatier, Toulouse, France 2005. http://www.irit.fr/~Philippe.Joly/Homepage_files/Theses/Bassem_Haidar.pdf

[2] B. Haidar, B. Chebaro. An opened distributed platform for automatic chaining of multimedia indexing tools. To appear in the International Journal of Reviews in Computing.

[3] KLIMT : V. Conan, I. Ferrané, P. Joly, C. Vasserot. KLIMT: Intermediations Technologies and Multimedia Indexing. In Third International Workshop on Content-Based Multimedia Indexing (CBMI'03), Rennes, France, september 2003. INRIA, 11-18.

[4] H. Mandviwala, S. Blackwell, and JM. Van Thong. A Scalable Multimedia Content Analysis and Indexing System Architecture. Third International Workshop on Content-Based Multimedia Indexing, Sept. 22-24, 2003, Rennes, France. 349-355.

[5] Mihaela Brut, Florence Sèdes and Ana-Maria Manzat. A Web Services Orchestration Solution for Semantic Multimedia Indexing and Retrieval. In Proc. of the Second International Workshop on Frontiers in Complex, Intelligent and Software Intensive Systems (FCISIS-2009), organized in conjunction with The Third International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009), IEEE Computer Society, pages 1187-1192, Fukuoka (Japan), March 2009.

[6] T zouvaras, V., Troncy, R., Pan, J., "Multimedia Annotation Interoperability Framework", W3C Technical Reports, Boston, 2007: http://www.w3.org/2005/Incubator/mmsem/XGR-interoperability/

[7] Gopi Kandaswamy, Liang Fang, Yi Huang, Satoshi Shirasuna, Suresh Marru, and Dennis Gannon. Building Web Services For Scientific Grid Applications. In IBM Journal of Research and Development, 2005.

[8] Sriram Krishnan, Brent Stearn, Karan Bhatia, Kim K. Baldridge, Wilfred W. Li, and Peter Arzberger, Opal: Simple Web Services Wrappers for Scientific Applications, 2006 IEEE International Conference on Web Services (ICWS 2006), Sep 18-22, Chicago

[9] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of service for workflows and web service processes. J. Web Semantics, 1(3):281–308, April 2004.

[10] J. Cardoso. "Quality of service and semantic composition of workflows". Ph.D Thesis, University of Georgia, 2002.

[11] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam and H. Chang. "QoS-Aware Middleware for Web Services Composition". IEEE Transactions on Software Engineering, pp311-327, Vol. 30, No.5, May 2004

[12] Fung C.K., Hung P.C.K., Linger R.C., Wang G. and Walton G.H. "A Service-Oriented Composition

Framework with QoS Management", International Journal of Web Services Research, 3(3), 2006, pp. 108-132.

[13] Menasce D.A. "QoS Issues in Web Services", IEEE Internet Computing, 6(6), 2002, pp. 72-75, ISSN: 1089-7801.

[14] G. Silver, A. Maduko, R. Jafri, J. A. Miller, and A. P. Sheth. Modeling and simulation of quality of service for composite web services. Proceedings of the 7-th World Multiconference

Vol. 13 No.2 March , 2010 pp (139 - 147)

on Systemics, Cybernetics and Informatics (SCI'03), 1:420–425, 2003.

[15] J. Xia, "Qos-based service composition," Proc. IEEE Int'l Conf. Computer Software and Applications Conference (COMPSAC'06). student paper, Chicago, IL, September 2006, pp. 359-361

[16] SOAP : Simple Object Access Protocol 1.1 http://www.w3.org/TR/soap/

[17] Apache Axis, SOAP implementation: http://xml.apache.org/axis

[18]DataHnadler : http://java.sun.com/j2ee/1.4/docs/api/javax/actvation /DataHandler.html

[19]MPEG-7: http://www.chiariglione.org/mpeg/standards/peg-7/mpeg-7.htm

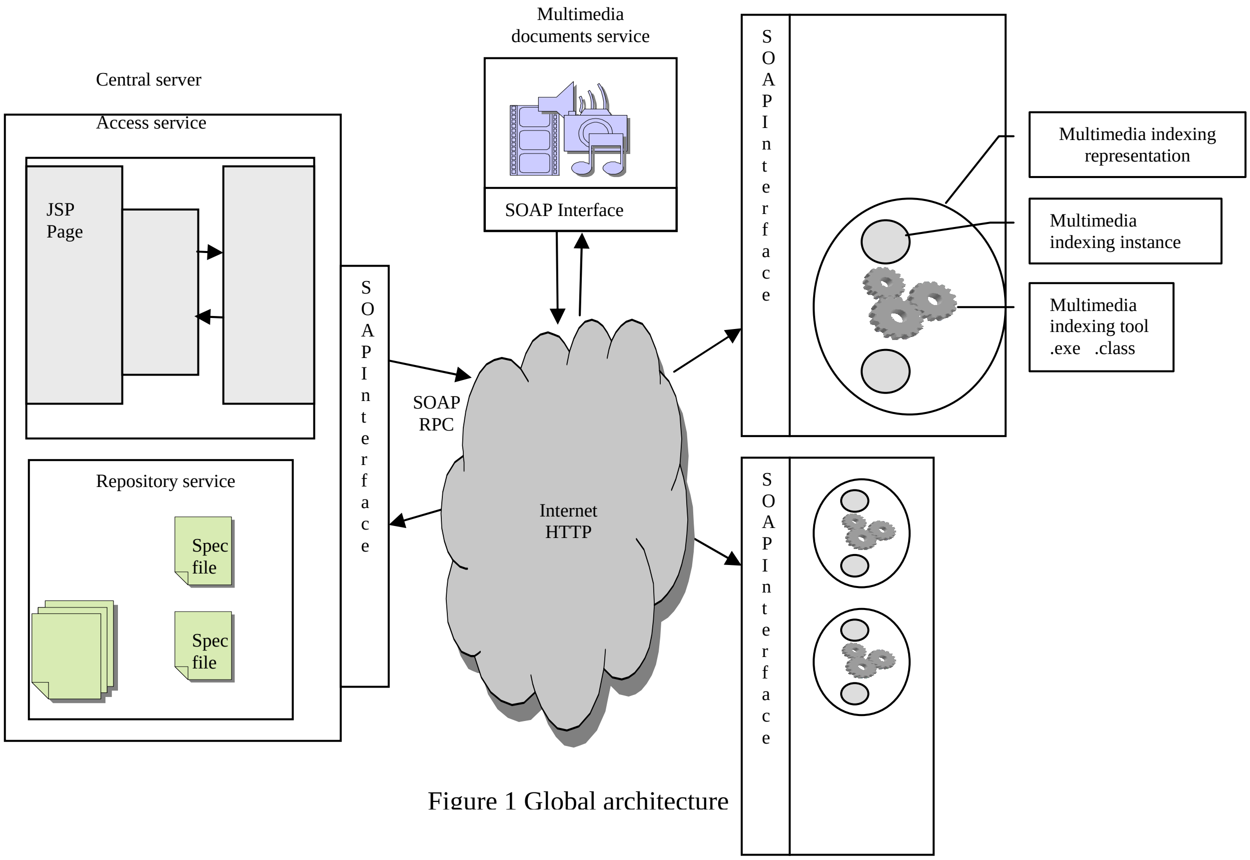

Figure 1 Global architecture

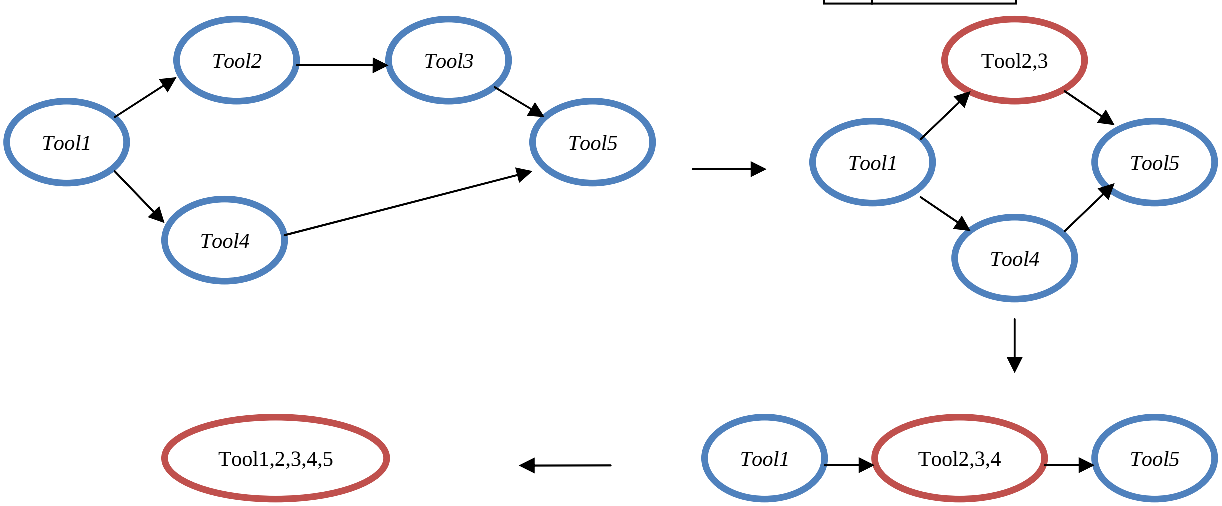

Figure 4

Vol. 13 No.2 March, 2010 pp (139 - 147)