Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 480 – 488

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

A NOVEL APPROAVH TO DISCOVER WEB SERVICES USING WSDL AND UDDI

Kanimozhi S[a], Kannana A[b], Selvamani K[c], Vijay Kumar A[d]

[a]Research Scholer, Anna University, Chennai,600025, India, [b] Professor, Anna University, Chennai,600025, India

[c]Assistant Professor, Anna University, Chennai,600025, India, [d]Research Scholer, Anna University, Chennai,600025, India

Abstract

Web service technology has become prominent in providing a dynamic integration and interaction of heterogeneous system, thereby to facilitate fast and efficient cooperation among the entities in cooperative communication environments. With the surge of service oriented architecture (SOA) and web services, service discovery has become increasingly crucial. In finding the appropriate web service discovery mechanism can retrieve relevant web services from the internet to serve a wide range of users such as service consumers, service developers, service deplorers, and service brokers. This paper aims in demonstrating the interpretation WSDL files to discover the required web services according to the user request and compare between WSDL and UDDI based discovery system. The result leverages in generating the WSDL from java, creating Tmodel of UDDI from WSDL, tokenizing the WSDL elements, comparing user requested service name with WSDL elements using MMA algorithm and generate WSDL element matrix in order to fully satisfy the user request in finding the appropriate web services according to WSDL files.

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

* Corresponding author. Tel.: +91-9840877521
*E-mail address:*selvamaniau@gmail.com

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.123

Keywords: WSDL, UDDI, SOA, MMA.

I. Introduction

Web services characteristics that posses, self- contained software modules which is loosely coupled environment accessed programmatically using a well as to Internet technology, and assembled dynamically to serve a particular function, as well as solve a specific problem and deliver a particular solution to a customer. The message communication between a service using the SOAP(Simple Object Access Protocol) is performed with communication protocols like HTTP, SMTP, or FTP. When the service receives the request message, The URI of service that discovers using UDDI(Universal Discovery and Description Integration) or WSDL(WEB Service Description Language).

Existing Web Service discovery methods are classified into the broad categories namely WSDL- based, Ontology-based, UDDI based. In ontology based method a notation have been made to reference to a domain ontology through the standard WSDL extension mechanism. Ontology based methods aims to provide 'semantically oriented' version of WSDL files in order to automate complicated tasks such as service composition.

This research paper describes the web service discovery using WSDL based method and UDDI based method. Normally web services are described using WSDL documents by WSDL documents while semantic web services use web Ontology language (OWL-S) as a description language. WSDL based discovery is most popular and supported by both industry and development tools. [1]WSDL based method is further divided into text based, Structure based and semantics based methods. Text based method is the most straight forward way to conduct web service discovery. The most widely used text based method is keyword matching built in the UDDI public registry.

In addition UDDI API allows developers to specify keywords of particular interest and then it returns a list of web services whose service description contains those keywords. The literal keyword method cannot tell the equivalence between Combo and Combination. The text used in this approach is extracted, analyzed and expanded directly from WSDL elements. This paper aims demonstrating how this can be achieved to get proper operation of web services and to discover the web services using WSDL based method as well as comparison between UDDI and WSDL based discovery method.

II. Literature Review

Chen Wu and Elizabeth Chang [2] indicated that public UDDI Business registry and the primary service discovery mechanism over the internet has been shut down permanently due to many factors. Hence, the most important web server discovery mechanism is missed from web service community.

Also, some Literature review justifies the necessity, of WSDL based discovery of web services by using WSDL based method.

Noh-Sam Park [3] et al explained about the consumer who search web services with UDDI and manually access the web services. But the UDDI search results only provides specifications for registered web services and not able to provide result that the user expects.

Khalid Elgazzar [4] point out that, more than 53% of UDDI Business Registry (UBR)are invalid where as 92% of web services crawled by search engines are valid and active. Also search engines partially match the search terms entered by the user with the web service name, location, business or tModel defined in web service description file to get the results back.

Lijie Wang [5] et al proved that large proportion of web services on the internet could not provide enough descriptions in their WSDL files. Hence it is necessary to enrich descriptions for public web services by extracting useful information and providing semantics by data mining technique. Pat. P. W. Chan and Michael R. Lyu [6] indicated the challenges for integrating the semantics of web services in automatic service composition as well as, semantics are captured through manual service composition.

Jan Hendrik Hausmann [7] described that OWL (Ontology Web language) is first step towards the creation of semantic web enabled Web Services. They have concentrated on the description of static information. Which contradicts the demand of a flexible description of innovative web service in the dynamic nature of ebusiness. When large scale web services are available an innovative dynamic structured integration is required.

Fang fang Liu [8] proposed an approach to find the similarity of services which is evaluated by the traditional measures such as Jaccard and Euclidian combined with WorldNet to increase the precision. But in practice, most text descriptions of published web services contain much useless information unrelated with the function of services, which hampers the application of this kind of approach.

Ning Gu[9] explained that SWORD(Software Ontology For Resource Description) is rule based expert system for web service composition. SWORD is set of tools for the composition of a class of web services including "information - providing ". SWORD can compose services automatically, but it identifies services only syntactically - that is, by their inputs and outputs.

Thomas Fischer [10] explained about recently WSMO-Lite(Web Service Modelling Ontology) for describing Web Services semantically as the next evolutionary step tiller SAWSDL (Semantic Annotations fin WSDI files). SAWSDL annotations with concrete service semantic service descriptions WSMO-Lite ontology is on one side lightweight and on the other side provides elements fir modelling functionality of web services. But WSMO-Lite does not provide modelling of input and output parameters explicitly and relies on their derivation from free variables in the formulas for precondition and effect.

III. WSDL

To make a Web service useful, a service consumer who has discovered a set of useful services must be able to determine invocation details of the services. 'This can he described by WSDL. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information_ The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint Related concrete endpoints are combined into abstract endpoints (services) [11]The abstract and concrete description of WSDI is given below.

A. Abstract Description

- ✓ Types: contain the platform- and language-independent data type definition.
- ✓ Messages: Contain input and output parameters of the service and describe different messages the service exchanges.
- ✓ Operations: represent a particular interaction with the service and describes the input, output, and exception messages possible during that interaction.
- ✓ Port Types: uses the messages section to describe function signatures (operation name, input and output parameters) and represents a set of operations supported by the service.

B. Concrete Description

- ✓ Bindings: specifies binding of each operation in the port types section. It associates the abstract descriptions of a port Type (i.e., a port Type's operations, messages, and data types) with a network protocol.
- ✓ Services. In addition to protocol-specific information, the WSDL document should also describe where the service is deployed. The association between a binding and the network address at which it can be found is defined by a port. The service element is a collection of ports, and a port describes a network location for a binding.

WSDL is extensible to allow to description of endpoints and their messages regardless of what message formats or network protocols are used to communicate. However, the only binding described in this document describes how to use WSDL in conjunction with SOAP 1.1,HTTP GET/POST, and MIME.

C. Generating WWL, from Java

Java programmers who have little interest in understanding the details of a WSDI document will be encouraged to know that there are tools to generate a WSDL document, given a Java remote interface. Current tools will, however, generate only WSDI, that &support an RPC style invocation for SOAP-based web services. The xrpee utility can be used to generate a WSDL document with SOAP-HTTP binding, given a Java remote interface, using the following command:

```
xrpcc -classpath %oclasspath% -server -keep -d <destination directory> <configuration xml file>
```

IV.UDDI

The ability to publish services in a UDDI registry requires the application used to publish the service interface definitions to understand WSDL. WSD1 4J is one mechanism that allows iut application to read and create "Fmodel of UDDI which contains URL of WSDL is given here.

// Read the WSDI service interface document Definition definition= WSDL Reader:readWSDL(null., wsdl URL);
Create a new tModel to be used to map the WSDL. Service interface
Tmodel tModel = new tModel();
OverviewDoc OverviewDoc = new OverviewDoc();
OverviewURL() OverviewURL = new OverviewURL(wsdlURL);
tModel.setOverviewDoc(oveniewDoc);

This is the last step in parsing WSDL, to create the appropriate UDDI entities. UDDI4J contains APIs that allows to publish, find, and bind to a Web service Because UDDI4J is open source, it comes with source code, JavaDoc, and several sample applications, It contains multiple APIs but the one most frequently used is the UDDIProxy class. Let us look at how UDDIProxy class interacts with a registry:

UDDIProxy proxy = new UDDIProxy();

An user who wanted to find all businesses that meet a specified criterion, such ,is companies that start with the name "Flute," would use the find- business method of the proxy similar to this:

Business List b1 = proxy,find _business ("Flute", null, 0);

The UDDI4J is used for querying a registry. The Java API for XML registries (JAXR) is another way to query the registry.

V.WSDL Based Discovery

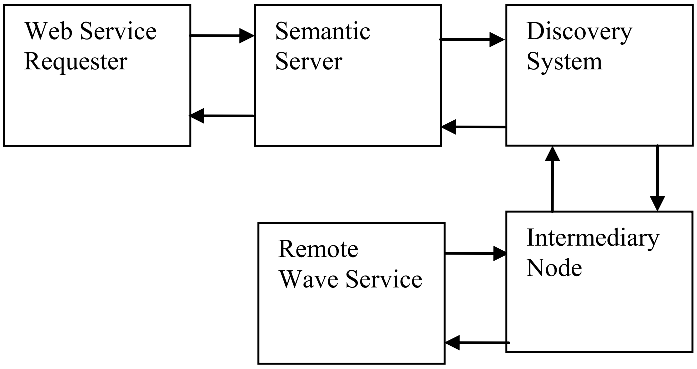
As explained in the literature survey additional efforts are required for discovery of web services and to get robust web service in low cost. Permanent shutdown of public UDDI has made UBR unavailable to start with. Therefore here instead of using MN it is better if we give query in the search engine to retrieve WSDL. The model given here in fig l gives overview of how a WSDL based discovery system satisfies the user request.

As shown in Fig I user requests -for a web service from web service requestor system. Semantics of the request is collected by semantic server which enriches the request with additional suitable semantics and proper request is directed to discovery system. Semantic server uses the well established model from Language processing as well as ontology of grammar to find the semantics of user request which is out of scope of the paper. Output of Semantic server contains keywords of user interest.

Discovery system uses these keywords to retrieve WSDL file of the web service, How to get suitable web service using WSDL processing system is explained in this paper. To interpret the service names from the user request, service name of user request is compared with domain knowledge to get suitable functionality. Domain knowledge is built as repository of database which holds service names of web services, those. belongs to social, commercial, educational web services, Me matching of requested service names(RSN — Requested Service Name) with stored repository of the service narnes(SSN Stored Service Name) is clone by using Knuth-Morris-Pratt (KMP) exact pattern-matching algorithm_ .After matching is found synonyms are considered to get different service names with same meaning, In the search engine like Google a query is given like.

Enriched RSN:WSDL

Customized commercial search engines generally provide large number of online web services in the form of WSDL files. But this requires more pre-processing in WSDI, file to analyze WSDI- portal like HTML pages and to infer relation between them and WSDL files in order to create WSDL focused discovery without using Google. Therefore it is necessary to collect proper WSDL and process WSDL, policy metadata files to get suitable web service. Schema and policy documents are used as metadata documents of the web services The WSDL, document of a service usually consists of ports, port Types, operations, input/output messages and other definitions to express its function The names of its components generally are concatenation of words which declares the function of the service such as " getAccountDetails " of Port Type element: It is function of the tokenizer to decompose the names into individual terms according to some rules. These rules are given here.



A. Tokenizing the WSDL elements

From the retrieved WSDL file, elements will be extracted such as “<service/>” and “<port type/>” Which are generally more important than others describing the overall capability of a Web -service. Once, <service> and <operation.> elements are extracted from WSDL, and expanded, these raw name tokens cannot be utilized directly due to various reasons such as the sublanguage patterns, machine-generated code, or programming conventions, etc. Therefore, they need to be converted to natural languages before being indexed using IR models. [121]Chen Wu and Elizabeth Chang used the most important step of linguistic technique is tokenization, by .which token is split into a sequence of smaller meaningful terms by a special tokenizer. For Example, the operation name "GetAccountDetail” shall be tokenized into three sequential terms — "Account", "Detail-. Seemingly. Such a tokenization appears as straightforward as to detect the lower case letters and upper case ones in the name token. However, the problem can become more complicated when considering name tokens such as "GetMyeBayServices-, "AUD2USIr,,ox downloadMP3Musie" For example, applying the simple capital letter rule for the first token gives the result "Get", "Mye", "Bay", and "Services”. This is not desirable since the company name "eBay" is mistakenly split into two different terms As a result, service retrieval on "eBay" cannot match the operation. Therefore in this research not only tokenization based on rule, making tokenized word meaningful by using domain dictionary such as WORDNET will be followed. The following table gives tokenization results.

Table1. Tokenisation results

WSDL Names	WSDL Tokens
GetAccountDetail	Get, Account, Detail
BillPayService	Bill, Pay, Service
Downloadi2Profile	Download, i2, Profile
FindecommerceWebsite	Find, ecommerce, Website

The tokenization is based on the Maximum Matching Algorithm (MM.), which has been widely used for Chinese segmentation studies. The basic idea of MMA is to use an external word list to verify the possible word tokens parsed out from the unsegmented text. The algorithm starts from the first character in a text and reads in one character in a time to from the 'character sequence' cs. After reading each character, it attempts to find in the word list the longest word W that starts with the character sequence cs. If W can be also found in the text (i.e. the remaining parts of W also matches the following characters read from text), the MMA marks a boundary at the end of the W and starts again from the following characters using the same longest matching strategy until reach the end of the character sequence. If none matching word found in the word list, the first character in the character sequence cs itself will be identified as a single word.

B. Match Making Algorithm (MMA)

The Keywords of user interest is compared with WSDL tokens by using MMA algorithm is given here. Tokenized parameters of <service> element of WSDL are stored in the array of WSDL_ Service_ Array. Comparing this WSDL_ Service_ Array with request Service name RSN_ Output using MMA returns exact or pug in or subsumes or fail which is given below.

```
InputMatch(RSN_Output.WSDL_ Service_ Array)
Global_degreeMatch=Exact
For each WSDL_ Service_ Record in the
WSDL_ Service_ Array do
{
Find WSDL_ Service_ Record such that
degreeMatch=DegreeofMatch(RSN_Output, WSDL_ Service_ Record)
if (degreeMatch is equal to fail) then return fail
if(degreeMatch<GlobalDegreeMatch)
globalDegrreeMatch=degreeMatch
retun sort (recordMatch);
}
DegreeOfMatch(Rsn_Output, WSDL_ Service)
If Rsn_Output= WSDL_ Service then return exact
If Rsn_Output is subclassOfWSDL_ Service then return [illegible]
If WSDL_ Service subsumes  Rsn_Output= WSDL_ Service then return plugin
If Rsn_Output subsumesWSDL_ Service then return subsumes
Otherwise return fail
```

If the above algorithm returns exact then it is found that <service> name is exactly matches with Requested Service Name. The next step is to compare Requested Service Name's operation, input, output, other parameters with other elements of WSDL, like <operation>, <input>, <output> etc using same MMA algorithm and generating WSDL weight matrix. For example from the RSN tokens if "get Account Detail" is obtained as functional description of the service then it is compared with <operation> name of following part of WSDl, after tokenization.

```
<porttype name="InfyBank">
<operation name="getAccountDetails" parameterOrder="String_1">
<input message="tns:InfyBank_
<output message="ins:InfyBank_ "/> </operation>
getAccountDetails"/>
getAccountDetailsResponse
</portType>
```

Here also MMA returns exact then other parameters of the Requested Service Name(RSN) will he compared with corresponding WSDL tokens like the input/output element of the <porttype> element by the Discovery system. If <input> as well as <output> element of <porttype> matches with user request then <port> element will be processed to extract <location> element.

The <location> element of WSDL is given below.

```
<portname="BanksFPort"
binding="tns:InfyBankbinding">
 <soap:address xmlns: wsdl="
http://schemes.xmlnsoap.org/wsdl">
```

<location="http://www.infybank.getAccounts </port>

The <location> element gives URL, of the physical location of the web service which is required by the discovery system. Once the location is found its non functional description such as availability, reliability, security, policy will be analyzed using. matadata of web services using WS-MetaDataExchange satisfaction of these factors be constructed as client request and this message will be redirected to the same physical address via SOAP etc intermediary nodes as shown In the Fig 1.

Routing and addressing information will be added to SOAP header by the intermediary SOAP nodes and SOAP message will be redirected through the several intermediary nodes according to address Le of location of the web service. The SOAP message will be received by the remote service, extracts body of the SOAP message, understands and executes the request and response message transferred hack via intermediary nodes to discovery system. Discovery system collects the results in to its dictionary which is required for further analysis. Semantic server will pose additional queries like context of the service to requester after getting results from discovery system. According to user response, if user gets satisfied with the results then the result will be forwarded to requester.

VI. WSDL Weight Matrix for Similar WSDL Files

The MMA algorithm may match to more than one <service> element of similar services Liiie [illegible]ang[5] says matching of <service> name in WSDI, is more important than other elements. As it is shown in the table-[illegible] the query given in the search engine will retrieve more than one WSDL, file for the same service. Therefore if more than one WSDL service name matches to the Requester's Service Name then it is required to analyze other parameters such as <operation>, <message>, <types> etc. Therefore in this research according to the importance of elements all main elements of WSDL, are given different weights according to their importance.

In this research as shown in the table 1 according to the importance of the elements <service> element weight is given 1, <operarion> is given 0.9,<message> given 0.8 and <types> given 0.7. In the WSDL weight matrix weight is also given for extent of match of WSDL elements with RSN parameters which is returned by MMA like exact or plugin or subsume matching factors(MF). Weight given here as I tbr exact match, 0.5 for plugin, 0.[illegible] for subsume and 0 for

If Requested Service Name is exactly matched with <service> element of WSDL by MMA then it returns 1.Weight factor of <Service> is already assigned as 1. The net weight (NWWSDL) of <service> element in the WSDL Matching matrix is stored by the formulae 1.

WSDL Record	Service Name	<service> WW[illegible]DL	<operation> WWSDL-0.9	<message> WWSDL-0.8	<types> WWSDL-0.7	Sum of NWSDL
WWSDL-1	GetAccountDetail	Exact 1*1=1	Plugin 0.5*0.9 =0.45	Exact 1*0.8=0.8	Exact 1*0.7=0.7	2.95
WWSDL-2	GetAccountDetailInfo	Exact [illegible]1=1	Subsume 0.25*0.9=0.225	Exact 1*0.8=0.8	Fail 0*0.7=0	2.025
WWSDL-3	AccountDet[illegible]	Plugin 0.75*1=0.75	Fail-0.0	Subsume 0.5*0.8=0.40	Plugin 0.75*0.7=0.525	1.675

Table 1. WSDL, Weight Matrix

$NWWSDL_{LJ}=WWSDL_j * MF_{i,j}$ (1)

N $SumNWSDL_1= \sum_{j=1}^{N}(NWWSDL_{jIJ})$ (2)

$NWWSDL_{lj}$ is Net Weight of [illegible] WSDL for j^{th} element where 1<I<N and I < J<N, $WWSDL_j$ is weight assigned to j_{th} element according to importance of elements where l < J<N MF_{ij} is Matching factor of RSN with i_{th} WSDL file for j_{th} element whereMF_1 gets exact or plugin or subsume If sum of weight of WSDL elements of a Particular WSD record (SumNWSDL as shown in formula (2) is more than other WSDL) records then that particular WSDL file is chosen for further analysis of availability and security aspects

For c g. Requester Service Name GetAccountDetail may matches with more than one services like GetPaymentDetaill, GetAccountDetailinfo, AccountDetail , then WSDL matching matrix is given in the table I. In the above table WSDL I matches properly with Requested Service Name_ 'therefore finding the sum of NWSDI, as given in the formula 2 gives more weight for WSDL1 than other WSDL records. Therefore WSDL1 record's <location>, element is extracted and quality factors like availability, security, cost is analyzed If the service is found suitable then request message for that service is formed and sent to the provider.

VII. Performance Results

As given in the section IV, the Model of UDDI contains an overview URL element that points to the location of the service interface definition (WSDL). Therefore to retrieve using UDDI in UDDI based method UDDI is to be available and WSDL, has to be retrieved from the <overviewURL> element to get service signature and other details. Therefore to find appropriate service information from UDDI and to get URI of WSDL and to retrieve WSW, takes more time in UDDI based method than directly retrieving WSDL using search engine.

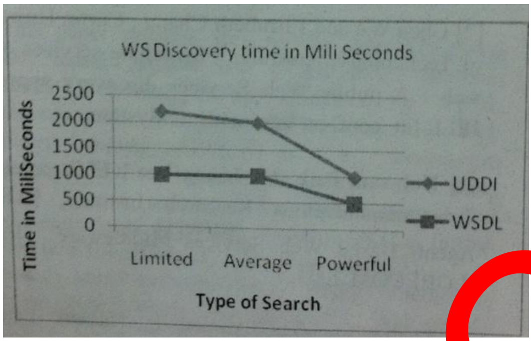


Figure2: Comparison between WSDL based discovery and UDDI Based Discovery.

VIII. Conclusion and Future Work

The proposed novel approaches to discover web services using WSDL and UDDI is implemented successfully. The various problem of ontology based method is solved by providing a suitable approach to discover web services.

This is achieved by the proposal of WSDL, based discovery system than UDDI and ontology based system. This proposed system is tested for different based queries, which results in finding appropriate services.

This proposed implemented work proved to be more robust and efficient approach to find suitable wave services.

But, it is necessary to provide robust web services to the user in less cost and high speed. Hence, future work in this direction cold be develop a knowledge base of discovery of web services to increase the speed of discovery using data mining technique.

IX. References

[1] Mohamed Gharzouli, Mahmoud Bouthida " A generic P2P Collaborative strategy for discovering and Composing Semantic Web services" IEEE 200 Fourth Int. Con!. on Internet and Web application and Services.

[2] Chen Wu and Elizabeth Chang , Curtin University of Technology, Australia " Searching services on the web A public web service discovery approach" IEEE fin. conf. on Internet Base System.

[3] Noh-sam Park, Gil-haeng Lee Electronics and Trelecommunications Research Institute, KOREA "Agent- Based Web Services Middleware" IEEE GLOBECOM 2003.

[4] Khalid Elgazzar, Ahmed E. Hassan, Patrick Martin , School of Computing, Queen's University, Canada " Clustering WSW., Documents to Bootstrap the Discovery of Web Services" 2010 IEEE International conference on web services

[5] Lijie Wang, Fei Lie, Liangjie Zhang, Ge Bin!! die „ Software institute, School of Electronic Engineering and Computer Science, Peking University, Beijing " Enriching Descriptions for public Web services using Information

Captured from Related Web pages on the Internet 2010 Fifth IEEE International Symposium on Service Oriented System Engineering.

[6] Pat. P.W. Chan and Michael R. Lyu Chinese University of Hong Kong , China ,"Dynamic Web Service Composition : A new approach in building reliable Web Service" 22nd IEEE Int. conf. on Advanced Information Networking and Application.

[7] Jan Hendrik Hausmann, Reiko Heckel, Marc Lohmann, University of paderborn " Model based Discovery of Web Services" IEEE International Conference on Web Services.

[8] Fangfang Liu, 'Yuliang, Shi, Jie Yu, -Fianhong Wang, J ingzhe Wu Measuring Similarity of Web Services Based on WSDI, 2010 IEEE International Conference on Web Services.

[9] Ning Gu junato Cui, Wei Ye, I laixun Wang, n Pei A system Framework for Web Service Semantic Automatic Orchestration" Granted by No 60473124.

[10] Thomas Fischer, Johaness Rubland, Friedrich Schiller University Jena "Towards Knowledge Discovery in the Semantic Web" , MIKWI 2010.

[11] James McGovern_ Sanicer Tyagi, Michael Stevens and Sunil Matthew" Java Web Services Architecture" a text-book ISBN :1558609008 Morgan Kaufmann Publishers 2003.

[12] "Searching services "on the Web": A public Web services discovery approach- Chen Wu and Elizabeth Chang Digital Ecosystems and Business Intelligence Institute Curtin University of Technology, Perth 68[illegible] Australia { Chen. Wu, Elizabeth. Chang }