Optimal Web Service Selection Using UML Profile

Rashmi Phalnikar, Devesh Jinwala

Abstract: Enterprises are more conscious of providing quality of services over the web for reasons of economy, reliability, interoperability and flexibility. Enterprise application relies on selection of the most appropriate service from several candidate services with similar capabilities provided by different service providers. The question is, on what basis the system chooses a service among several candidates. This paper proposes a model that makes an automatic selection of best service and detects the variance between the non-functional requirements of the users and service qualifications. In this paper, we describe our approach aimed to detect conflicts between user requirements and the service specifications of the service provider. Our work proposes to detect these conflicts using Ontology and UML profiles to achieve better performance and avoid unpredictable state of the system. We suggest use of UML extensions and domain Ontology to detect NFR conflicts between the client's requirements and service specifications.

Keyword: SOA, Ontology, Web Service Selection, UML Profiles, Conflict Detection

I. INTRODUCTION

The growing demand to increase the flexibility of the IT infrastructure to support rapidly evolving business needs, has led to a rising interest in Service-Oriented Architectures (SOA). At the heart of SOA is the concept of a "service" – a network enabled application that allows clients to connect by exchanging simple messages. Services implemented using any technology define a standard set of messages which clients will use to communicate with the service. In this way implementation details (and complexity) of the service are hidden from the client. The interoperability problem may have been addressed by SOA but an implementation framework must be in place to make it work particularly over the World Wide Web. Web Services is an effort to standardize the exchange of messages between clients and web applications.

The wide variety of services offered currently to perform a specific task to a user, indeed makes the task of appropriate service selection difficult. Hence, it is desirable to have due system support in the eventual selection of appropriate services for the user. In doing so, it is necessary to consider Quality of Service (QoS) parameters. The objective is to maximize the utility function under the end-to-end QoS constraints.

A service has two components viz. the functional requirement and a non-functional requirement (NFR) that represent an important facet of service descriptions in a SOA.

Web Service definitions are expressed in XML by use of the Web Service Definition Language (WSDL). WSDL description only addresses the functional aspects of a web service without containing any useful description of NFR or QoS characteristics.

The proposed approach has the following advantages:

- In our approach it is possible to quantify how NFRs affect the system's working. Our process combines evaluation and selection activities rather than only address selection issues. The user can decide whether this is acceptable or not.
- The approach proposed makes model reusable and applicable to a wide range of NFR as long as a Domain Ontology is in place. This is logical as compared to ranking all kinds of services by using the same predefined criteria and not considering the different attributes that occur with specific services.

We propose a conflict detection methodology to overcome these drawbacks and develop a model for help identify a web service which is closest to the user requirements. This will not affect the working of the system negatively. The proposed method will identify conflicting NFRs using UML Profile specifications and Domain Ontology.

II. THEORETICAL BACKGROUND

A. *UML Profile Diagrams*

Profile diagram is structure diagram which describes lightweight extension mechanism to UML by defining custom stereotypes, tagged values, and constraints.[4] Profiles allow adaptation of the UML metamodel for different:

- Platforms (eg. J2EE or .NET), or
- Domains (real-time or business process).

One way of using UML Profile is by creating and defining a domain-specific viewpoint that allows the model to be interpreted from different points of views. UML profile The ability to dynamically apply and un-apply a UML profile without affecting the underlying

DOI: 10.5176_2010-2283_2.1.157

model is crucial to the second type of profile usage, because it allows the same model to be viewed from different viewpoints. The profiles mechanism:

- Does not allow to modify existing metamodels or to create a new one
- Allows adaptation or customization of an existing metamodel with constructs that are specific to a particular domain, platform, or method.
- It is not possible to take away any of the constraints that apply to a metamodel.
- Can be dynamically combined so that several profiles will be applied at the same time on the same model.

B. Web Services

A web service is defined as [1] a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-readable format (WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Web Services architecture then requires three fundamental operations: publish, find, and bind. In Web Services technology, Web Services described in WSDL are advertised in UDDI (Universal Description, Discovery and Integration) registries. UDDI provides only keyword-based discovery (e.g. service category or provider name) . Thus service discovery is restricted to simple keyword-based category and attribute matching. SOAP, defined as Simple Object Access Protocol, in XML is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks.

Contemporary Web Service Discovery algorithms allow the discovery and selection of web services based on syntax and semantics of the service. We however argue that this is inadequate, and to achieve best performance NFR need to be considered. These considerations include for e.g. the range of time period during which a service is relevant, quality of service, delivery policy, regulatory constraints, payment methods, etc. Other than these, customers may have some constraints or conditions to meet, such as preference in the payment methods or delivery dates. Some examples include "Pay a supplier invoice only if it has been approved." "Only good customers may obtain credit orders." "Overdue invoices occur 30 days after statement." "Many payments can be made per invoice." "Only one invoice should be generated for one order." "Credit balance should be greater than or equal to order value to accept order, otherwise reject the order."

As per Chun et.al. [23] the level of compatibility among services can be defined by three levels of rules - syntactic, semantic and policy.

- Syntactic (operational) Rules: depends on the preconditions, input, and output requirements.
- Semantic Rules: Are rules for standard business practices that require domain knowledge.
- Policies: rules that are restricted by the policy compatibility requirements.

Our work will handle semantic rules that are unique to a particular domain.

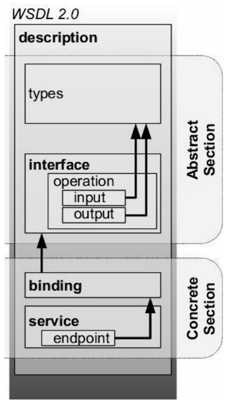

Fig. 1: Concepts defined by WSDL 2.0

The current version of WSDL is WSDL 2.0.A WSDL description of a web service provides a machine-readable description of how the service can be called, what parameters it expects and what data structures it returns. A WSDL document can also contain other elements, like extension elements, and a service element that makes it possible to group together the definitions of several web services in one single WSDL document.Fig. 1 depicts the concepts defined by WSDL. The WSDL document defines services as collections of network endpoints, or ports. The abstract definitions include:

- messages, which are abstract descriptions of the data being exchanged, and
- Port types which are abstract collections of operations are allowed.

The concrete protocol and data format specifications are for a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service.

As web services gains popularity, research works address implementation and execution issues. WSDL has its advantages as a universal representation and exchange format, but it can be difficult to understand and to write for non-XML experts. A standard graphical modelling language should be employed in combination with an XML-based representation. As UML is already used as a Process Modeling Language and it has some useful features like standardized, graphical user interface allows to model different views of a system, it naturally become the first choice.

They can be transformed to directly executable composition specifications; and they are independent of the executable composition languages.

As mentioned earlier, WSDL describes three fundamental properties of the Web service: what a service does (the operation that the service provides), how a service is acceded (details of the data formats and protocols necessary to access the service's operation) and where a service is located (details of protocols-specific network address, such as URL).

For conversion from WSDL to UML the specifier has to first import the Web services he wants to match his request with by providing their WSDL file's URL. From these WSDLs, the UML diagram is generated by representing the interfaces of the Web services involved as well as the complex data types they use. The modeling experience can be broadly divided into creation of two WSDL partitions:

i. Platform-Independent Model: represents the abstract portion of WSDL. It models Definitions. Service. Port type(s). Messages. Parts. Part type(s).

ii. Platform-Specific Model**:** bindings section of WSDL. It models Service. Ports. Binding.

The necessary and sufficient UML diagrams required to represent the two WSDL partitions are as follows:

- Platform Independent Model – Class View
- Platform-Specific Model - Class, Component and Deployment View

Much of the work done use an UML extension for Web service representation based on WSDL. The extension gives a UML notation that allows representing a Web service and, also facilitates the automatic generation of WSDL description of a Web service from an UML diagram. We are interested in generation of UML diagrams from WSDL only.

C. Domain Ontology

Ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. Probably the most condensed definition originates from T. Gruber [3]:

"An ontology is a formal explicit specification of a shared conceptualization."

Domain Ontologies are descriptions of particular subject or domain areas. They are the "world views" by which organizations, communities or enterprises describe the concepts in their domain, the relationships between those concepts, and the instances or individuals that are the actual things that populate that structure. Thus, domain ontologies are the basic bread-and-butter descriptive structures for real-world applications of ontologies. Developing Ontology has many benefits:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

Specifically lightweight ontologies are defined as more hierarchical or classificatory in nature. Lightweight ontologies are often designed to represent relationships between concepts. They have not too many or not too complicated predicates (relationships). As relationships are added and the complexities of the world get further captured, ontologies migrate from the lightweight to the "heavyweight" end of the spectrum.

The main components of ontology are concepts, relations, instances and axioms. A *concept* represents a set or class of entities or `things' within a domain. Concepts fall into two kinds:

- Primitive concepts are those which only have necessary conditions (in terms of their properties) for membership of the class.
- Defined concepts are those whose description is both necessary and sufficient for a thing to be a member of the class.

Relations describe the interactions between concepts or a concept's properties.

In general terms, the ontology development can be divided into two main phases: specification and conceptualization. The goal of the specification phase is to acquire informal knowledge about the domain. The goal of the conceptualization phase is to organize and structure this knowledge using external representations that are independent of the implementation languages and environments.

In practical terms, developing ontology includes:

- Defining classes in the ontology,
- Arranging the classes in a taxonomic (subclass–superclass) hierarchy.

Defining slots and describing allowed values for these slots, filling instance values for slots.

III. RELATED WORK

Even though our work is inspired by the above mentioned references, our proposal of use of Ontology and UML profile for conflict detection between service users and provider in SOA environment is unique and we expect it to give good results by improving QoS significantly by reducing ambiguity and promoting reuse.

As mentioned earlier, current approaches for service selection do not provide an automated service identification framework, and most service identification methodologies ignore important aspects such as performance metrics and conformance of the identified services with SOA principles.

To the best of our knowledge little study has been done on the manner and degree in which conflicting NFRs can behave. To overcome this gap we propose to define a UML profile for NFRs using UML extension mechanisms, specify the semantics of stereotypes and tagged values for mapping to a process and then derive rules to detect inconsistencies.

i. Qos Based Web Service Selection:

QoS based service selection aims at finding the best service which satisfies the user requirements. SLA between the consumer and Web service defines the QoS agreement. Different methods have been suggested and studied. Ran [5] proposed an extended service discovery model containing the traditional components: service provider, service consumer and UDDI registry, along with a new component called a Certifier. Certifier verifies the QoS of a web service before its registration. However, it lacks support for the dynamism of web services, the work fails to illustrate the quantifiable measurements as it simply assumes that all measured values are available somewhere.

Singhera [6] and Rajendran et.al [7] proposed UDDI extension to support run-time collection of data/information related to non-functional characteristics of web services. However these approaches do not provide guarantee as to the accuracy of the QoS values over time or having up-to-date QoS information. Zeng et al. [8] present a middle ware platform that enables the quality-driven composition of Web services. The QoS is evaluated by means of an extensible multidimensional model, and the selection of Web services is performed in such a way as to optimize the composite service's QoS.

Much of the work done earlier highlights the importance of QoS and attempts to incorporate NFRs into the service description. To the best of our knowledge no study mentions the way NFRs conflict and makes system inconsistent. This is the focus of our work which is unique as it attempts to study the consequence of conflicting NFRs and uses UML Profile and Ontology concepts.

ii. UML Profiles And NFR

UML has been used for related fields like Impact Analysis, Conflict Detection and Version Control wherein the models have been compared or differentiated. Consistency rules for UML describe conditions that an UML model must satisfy for it to be considered a valid UML model using well formedness rules, coherence between different diagrams, and even coherence between different models.

Initial work on NFRs by Chung [9] describes a NFRs framework that provides a detailed process for refining a NFR from a very high-level abstraction to a design-level decision. This refinement is done through an AND/OR graph where the leaf nodes represent the design decisions which need to be implemented for achieving a particular NFRs. Selic [10] describes the most important innovations in UML 2.0 and a precise definition of profiles and stereotypes. Using formal rules for writing OCL constraints attached to stereotypes, they explained the semantics of applying (and un-applying) profiles to UML models .The rules for an XMI representation of profiles and their contents were defined.

Beek et.al [11] formulate conflict detection in UMC which is a model checker built to analyze UML state machines In UMC a state machine diagram is associated to the notion of class, while a system's configuration is defined by a set of objects (active class instances). Consistency is maintained in UML models by Straten [12] by use of an extension of the UML metamodel and Description logic (DL). They use three kinds of UML diagrams: class diagrams, sequence diagrams and state diagrams. Straten defines a two dimensional inconsistency conflict table wherein Conflicts can occur at the Model level, between the Model and an instance.

Work based on descriptive logic by Wageman [13] defines two types of inconsistency: horizontal and evolution consistency. They have discussed model-model conflict, model-instance conflict and instance-instance conflict in class, sequence and state diagrams between different versions of UML diagram. Husseini et. al. [14] describes a UML profile called UMLintr (UML for intrusion specifications) that allows developers to specify intrusions using UML notations. This approach also helps to avoid conflicting (e.g., security vs. usability), ambiguous, and redundant requirements.

Egyed [15] presents an approach for quickly, correctly and automatically deciding what consistency rules to evaluate when a model changes. Briand [16] detects changes in UML models and analyses the impact due to the change. A measure of distance between a changed element and potentially impacted elements is calculated. The initial steps include checking of the well formedness rules given by OMG [17]. They checked about 120 of the rules defined.

Suppakal et.al. [19] propose an integrating modeling language by extending UML with the NFRs Framework using UML Profile. They define a metamodel to represent the concepts in the NFRs Framework and they identify the extension points for integrating the two notations.

iii. From WSDL to UML transformation

UML has been considered to describe Web services composition. An extension given by Dumez et.al [21] suggests that both the WSDL file of a Web service and its UML-S class diagram contain its name its methods and the complex types involved. Skogan et.al. [22] describes an approach using UML activity diagrams. They provide a way to model the coordination and the sequencing of the interactions between Web services. Jiang [23] proposed UML-based profiles to define structural rules of WSDL documents for WSDL descriptions. These profiles can be used to guide the user in designing correct and Basic Profile compliant WSDL descriptions and to check the validity of existing WSDL descriptions.

Ha et.al. [22] adds ontology-based framework to Web service generation system. They combine ontology framework and Web service generation dynamically.

IV. THE PROPOSED APPROACH

Currently, the process of web discovery and service selection is based on the user making the decisions as to which service is appropriate for purpose. In addition, matching is mostly based on functional requirements while non-functional properties are not considered. Different web services with different QoS requirement will bring competitive edge for service provider. To provide a better QoS it is necessary to identify an appropriate web service that satisfies requirement completely.

Moreover, the increasing availability of Web services that offer similar functionalities with different characteristics increases the need for more sophisticated discovery and selection processes to match user requests. This is where we feel that the proposed work will be useful.

To overcome these drawbacks and develop a model for help identify a web service which is closest to the user requirements and does not negatively affect the working of the system, we propose a model that identify conflicting NFRs using UML Profile specifications and Domain Ontology.

The flowchart for this work is given below in Fig. 2:

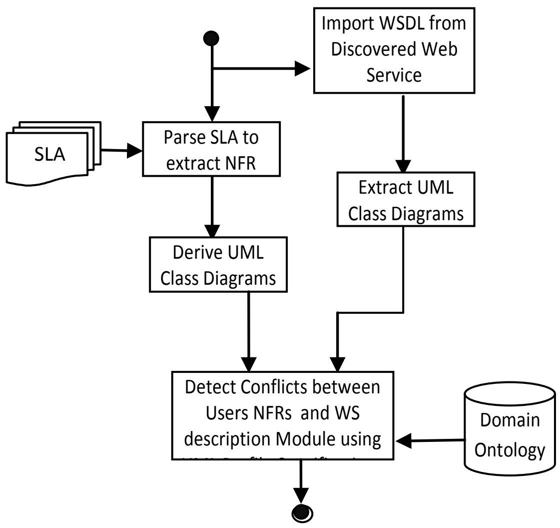

Fig.2 Flowchart for Proposed work

The proposed work can be divided into three modules at this stage:

i. Extracting User's NFRs from the Service Level Agreement (in textual format) into UML diagram.
ii. The web service's WSDL is imported by using its URL. From here a profile framework developed should generate a class diagram that presents the interface of Web service and the complex data. Much of the research we studied uses the conversion of UML diagram to WSDL for Web service composition. We were not able to find substantial work that uses the benefit of illustrating WSDL using UML concepts.
iii. Having created these two UML diagram we could apply them as profile for the application package. The crucial part of the work will be in defining these profiles and applying them in such a way that the conflicts are detected. These extension mechanisms allow refining standard semantics only in an additive manner so that they cannot contradict themselves. Object Constraint Language (OCL) will be the next step. UML Profiles have been explored for model version control and impact analysis .However our work is unique in the sense that we should be able to make a judgement of the effect on the system of conflicts.
iv. Use of Domain Ontology is restricted to explanation of domain for clearer understanding and mapping. Experience with defining profiles has indicated that it is best that the initial domain model is not ambiguous .We could make use of an existing Ontology rather than develop a completely new one.

Using our approach, new policies can be added at operation time and checked for consistency before actual insertion of policy set.

V. CONCLUSION AND FUTURE WORK

The quality-based selection of Web Service Selection is an active research topic in the area of dynamic service discovery. From the papers surveyed and study of recent work done , we can conclude that even though much work has been done on web service discovery and composition, due consideration is not given to conflicting NFRs between service providers and service users.

This work proposes a way in which profiles can be integrated into the UML Metamodel and can be used to detect contradictory NFRs that can make the system inconsistent. The associations between the UML models, profiles and the user requirements will help to establish the importance of NFRs in web service selection. This will lead to a maintainable system with better performance. Also, this approach should allow the study of outcome of mismatched NFRs, and notion of inconsistency.

We propose a method wherein quality characteristics from the WSDL description is retrieved and mapped with user's NFRs. The user's NFRs or policies can be described using UML diagrams which are derived using NLP techniques. This QoS-based model is expected to enhance the WS selection process and elevates the effectiveness of the delivered services as certain constraints are fulfilled.

To the best of our knowledge this proposed method in mapping of NFRs in web service domain is a novel approach and can provide good results.

Future steps indicating next research attempts and directions in the area include:

- Designing of a mechanism that derives UML diagrams, one from the textual requirements provided by the user and the other from the WSDL description of web service selected.
- Exploring the application of UML profile and extensibility mechanism in detecting factors that may impact other elements in the model.
- The UML profiles will treat nonfunctional requirements as first-class elements. Further we could trace the consequence of a changed requirement in the UML Profiles.

REFERENCES

[1]. Latha Srinivasan and Jem Treadwell , "*An Overview of Service-oriented Architecture, Web Services and Grid Computing*" by HP Software Global Business Unit ,November 3, 2005

[2]. Marc Oriol Hilari, "Quality of Service (QoS)in SOA Systems" Masters in Computers, Universitat Politècnica de Catalunya, Barcelona, 2009, Print

[3]. Thomas R. Gruber, A Translation Approach to Portable Ontology Specification. Knowledge Acquisition 5: 1993,Pps. 199-220.

[4]. Martin Fowler, "*UML distilled: a brief guide to the standard object modeling language",* Adddison Wesley , USA , 2004

[5]. Shuping Ran. 2003. "A model for web services discovery with QoS" In SIGecom Exch. 4, 2003, Pps. 1-10.

[6]. Zafar U. Singhera and Abad Ali Shah., "Extended web services framework to meet non-functional requirements" In *Proc. Of Sixth international conference on Web engineering* (ICWE '06). ACM, Article 21 .

[7]. T.Rajendran Dr.P.Balasubramanie Resmi Cherian , "An Efficient WS-QoS Broker Based Architecture for Web Services Selection" , International Journal of Computer Applications, 2010, Volume 1 – No. 9, Pps. 79-84

[8]. Liangzhao Zeng, Boualem Benatallah , Anne H. H. Ngu and Marlon Dumas and Jayant Kalagnanam and Henry Chang," QoS-aware middleware for Web services composition", Journal of IEEE Transactions on Software Engineering, 2004,Vol. 30, Pps. 311--327

[9]. Lawrence Chung and Brian Nixon,"Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach",In Proc. Of 17th ICSE,-1995, Pps. 25-37.

[10]. Bran Selic , "A Systematic Approach to Domain-Specific Language Design Using UML",In Proc. Of 10th IEEE International Symposium on Object and Component Oriented Real Time Distributed Computing ISORC07 (2007) , Pps. 2-9

[11]. H. ter Beek, S. Gnesi, C. Montangero, and L. Semini, *"Detecting Policy Conflicts by Model Checking UML State Machines"*. In Proc. Of Feature Interactions in Software and Communication Systems X ,2009,Pps. 59 - 74.

[12]. Tom Mens, Ragnhild Van Der Straeten, Jocelyn Simmonds," Using Description Logic to Maintain Consistency between UML Models Management, In Proc. Of 6th International Conference, The Unified Modeling Language, Modeling Languages and Applications, USA, 2003, Volume: 2863, , Pps 326-34

[13]. Jocelyn Paola Simmonds Wagemann, Consistency Maintenance Of Uml Models With Description Logics, Master of Science in Computer Science, Vrije Universiteit Brussel – Belgium, 2003

[14]. Mohammed Hussein and Mohammad Zulkernine, "UMLintr: A UML Profile for Specifying Intrusions**",** In Proc. Of 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06) ,Pps. .279~288

[15]. Alexander Egyed, "Instant consistency is checking for the UML" In *Proc. Of 28th international conference on Software engineering* ICSE , 2006 ACM, Pps. 381-390

[16]. L. C. Briand, Y. Labiche, and L. O'Sullivan, "Impact Analysis and Change Management of UML Models" In *Proc.of International Conference on Software Maintenance* ICSM , 2003. Pps. 256-261.

[17]. Object Management Group (OMG), "OMG Unified Modeling Language (OMG UML) Superstructure", V2.1.2, November 2007.

[18]. Sam Supakkul, "A UML profile for goal-oriented and use casedriven representation of NFRs and FRs ",In Proc. of the 3rd International Conference on Software Engineering Research, Management and Applications, 2005, Pps. 112-121.

[19]. Christophe Dumez, Ahmed Nait-sidi-moh, Jaafar Gaber, Maxime Wack, "Modeling and Specification of Web Services Composition Using UML-S," In Proc. Of International Conference on Next Generation Web Services Practices, 2008, Pps. 15-20.

[20]. David Skogan, Roy Grønmo, Ida Solheim ," Web Service Composition in UML" **,** In Proc. Of 8th International Enterprise Distributed Object Computing Conference, 2004 ,Pps. 39-45

[21]. Juanjuan Jiang, Tarja Syst, "UML-Based Modeling and Validity Checking of Web Service Descriptions," In Proc. Of IEEE International Conference on Web Services, 2005, Pps. 453-460.

[22]. Yan Ha and Roger Lee, "Semantic Web Service Modeling using UML for e-business environment", In *Proc. Of Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing* 2006, Pps. 368-374.

[23]. Soon Ae Chun, Vijayalakshmi Atluri, and Nabil R. Adam."Using Semantics for Policy-Based Web Service Composition", Journal of *Distributed Parallel Databases*, 2005, Volume 18, Issue 1 ,Pps. 37-64.

Rashmi Phalnikar is a research scholar at Sardar Vallabhai NIT Surat, India. Her research interests include Requirement engineering, conflicting requirements and UML Modelling. She completed her engineering from BV COE , Pune , India She is a member of the ACM. She can be contacted at Anant, 38/27 Erandvana , Pune 411004. India; rashmiphalnikar@yahoo.co.in.

Devesh Jinwala is Associate Professor in Computer Engineering at the Department of Computer Engineering, Sardar Vallabhbhai NIT SVNIT, Surat, India. He has a Ph D in Computer Engineering. His research interest include Information Security & Privacy in Wireless Sensor Networks, Software Reliability and Using Ontologies in Software Specifications.He can be contacted at dcj@svnit.ac.in.