Chapter 3

Related work

In this chapter, the existing work related to Web services research and application for Healthcare sector is presented. Section 3.1 discusses SOAP-based services discovery, selection and composition; Section 3.2 presents SOAP-based services discovery, selection and composition; Section 3.3 presents Cloud services discovery, selection and composition; Section 3.4 summarizes the contributions made by various researchers.

3.1 SOAP-based Web services discovery, selection and composition

This section is divided into two subsections: first subsection describes the existing approaches related with my work, while second subsection presents the application of Web services-based approaches in Healthcare sector.

3.1.1 Approaches for Web services discovery, selection and composition

Among the related approaches, I have considered the work proposed by (Yu and Bouguet-taya), (Zunino and Campo), (Mukhopadhyay and Chougule), (Ngan and Kanagasabai), (Dustdar and Schreiner), (Rao and Su), (Alamri, Eid, and El Saddik) for a comprehensive study on approaches for service discovery, selection and composition, specially semantics and QoS-based approaches.
An integrated approach proposed by (Yang and Papazoglou) towards service composition life cycle, which covers the service discovery, composition and selection of composed services without considering Semantic Web and QoS aspects.

Service discovery and composition issues of Web services in the proposed extended SOA (xSOA) architecture have been discussed by (Papazoglou and Van Den Heuvel). They have presented the concept of Semantic Web services to automate service discovery and composition process.

(i.) **Workflow-based Approaches**

The interleaving Web service discovery and composition approach was addressed in (Lassila and Dixit), by considering simple workflows, where Web services have one input and one output parameter. In this work, the Web service composition plan is restricted to a sequence of limited Web services corresponding to a linear workflow. In my framework, I propose a composition plan of services with multiple inputs and outputs, and also consider the other phases of the life-cycle of the service composition process, such as discovery and selection.

An algorithm for automatic composition of services was presented in (Zhang, Arpinar, and Aleman-Meza). The service composition is considered as a directed graph, where nodes are linked by the semantic matching compatibility (Exact, Subsume, PlugIn, Disjoint) between input and output parameters. Based on this graph, the shortest sequence of Web services from the initial requirements to the goal can be determined. This approach computes the best composition according to the semantic similarity of output and input parameters of Web services, but it does not consider any non-functional properties of the service composition.

An approach for Semantic Web Service composition has been presented by (Sirin, Hendler, and Parsia), which supports requesters to select Web services during each step in the composition process, and to create workflow specifications to link them. The discovery process consists of subsumption-based matching to filter services that can provide a concept as input for the current service at each step. This approach is semi-automated and totally dependent on human interaction. Therefore, it may be
infeasible for large service repositories. Moreover, this approach does not consider QoS constraints.

The authors (Yu, Zhang, and Lin) proposed heuristic algorithms to find a near-to-optimal solution. The authors propose two models for the QoS-based service composition: 1) a combinatorial model and 2) a graph model. A heuristic algorithm is developed for each model. Despite the significant improvement of these algorithms, both algorithms do not scale with respect to an increasing number of Web services and remain out of the real-time requirements.

(ii.) Genetic Algorithm-based Approaches

Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are the representatives of Evolutionary Algorithm (EA). The PSO is a swarm intelligence technique first proposed in (Eberhart and Kennedy). By applying PSO to QoS-aware Web service composition, Web services would be viewed as particles in a N-dimension search space, which determined by the number of QoS indicators and constraints (Huang et al.). The author apply an improved PSO named as IDPSO to QoS-aware WSC by altering step size factor under different situation. The step size would be larger, when particles are far from optimal target; on the contrary, the step size would be smaller when particles approximate optimal target.

Genetic Algorithm is another popular technique that has been widely used in QoS-aware Web services composition (Weise et al.), (Ma, Wang, and Zhang), (Canfora et al.), (Fanjiang et al.), (Modi and Garg). The idea of GA inspired from the biological evolution, where duplication, crossover, mutation and selection are key functions. In GA-based QoS-aware Web services composition, solutions are encoded as chromosomes, where each code represents a service involved. In each iteration, the quality of solutions is evaluated and the top chromosomes chosen to perform crossover and mutation until the fitness requirement is reached. GA-based approach provides optimized solution with assurance, but it is a time consuming technique, so it is not appropriate to apply the approach for dynamic composition process. (Tang and Ai) has put forward a hybrid GA for seeking services with global optimal QoS value.
and least violation of constraints. A fitness function, punishment function and local optimizer are used to realize the optimization of population.

(iii.) **AI planning-based Approaches**

AI planning techniques have been proposed as a way to automate Web services composition ([Carman, Serafini, and Traverso](#), [McDermott](#), [Sirin, Hendler, and Parsia](#), [Fadel](#), ([McIlraith and Son](#), [McIlraith and Son](#), [McDermott](#)) extended PDDL (Planning Domain Definition Language) ([McDermott et al.](#)) to deal with Web service composition. It was shown that the Optop (Opt-based total-order) planner (an extension to Unpop) could handle simple Web service composition. Even though McDermott proved that the estimated-regression planner is suitable for Web service composition, issues of dealing with non-determinism, partial observability and generation of conditional and iterative behaviors were not addressed.

SHOP2 ([Sirin et al.](#)) is a domain-independent planning system based on ordered task composition, which can greatly reduce the complexity of the planning process by avoiding some task-interaction issues. One of the shortcomings of SHOP2 is the inability to deal with non-deterministic actions. To resolve this problem, ND-SHOP2 planner ([Kuter and Nau](#)), a non-deterministic version of SHOP2, was developed. The main disadvantage of these approaches is that certain decomposition rules need to be encoded in advance with the help of process ontology. In order for decomposition rules, good knowledge of the domain is required.

([McIlraith and Son](#) and [McIlraith and Son](#)) extended and adapted Golog for automatic Web service composition. They use formal methods to define, characterize and compute the preconditions and conditional effects for complex action. Service composition is done via the use of general templates that are modified based on user preferences. Later, they applied a modified version of ConGolog (Concurrent Golog) to the Web service composition problem ([McIlraith and Son](#)). Extensions were added to the ConGolog interpreter, so that it has the ability to implement sensing actions (for information gathering) as external function calls.

([Ponnekanti and Fox](#)) presented SWORD for Web service composition, which uses
a rule-based expert system considering the Rete algorithm (Forgy). SWORD faces several limitations, such as it does not support automatic service discovery.

Automated Web service composition was performed through planning as model checking, by modifying the Model-Based Planning (MBP) system (Pistore et al.). MBP receives as input Web services, represented as abstract processes in BPEL4WS, and a given goal process. It generates a description of the required composite service in BPEL4WS. This approach copes with issues, such as partial observability. Semantic information was not considered during composition.

A declarative approach (Gomadam et al.) was presented to compose Web services, using planning dynamically. This approach overcomes the issue of syntactic heterogeneities at the data and the functional level, through mediation based matching, describing semantic information in SAWSDL, and uses an extension of the Graph-Plan to achieve a solution as a BPEL.

(Vuković, Kotsovinos, and Robinson) presented a framework that employs AI planning to rapidly assemble applications on-demand from individual services, based on context and user goals. The composition process utilizes the available services and the context, to create a composite service that satisfies the user’s task intention.

(Hatzi et al.) proposed an integrated approach for automated semantic web service discovery and composition using AI planning techniques. In this approach, OWL-S Web service descriptions are transformed into a planning problem using PDDL. The implementation of approach is demonstrated by the development of the PORSCE II and VLEPPO systems.

(iv.) Semantic Web-based Approaches

(Da Silva, Pires, and Van Sinderen) developed DynamiCoS (Dynamic Composition of Services) framework to address all the steps and stakeholders of the dynamic service composition life-cycle. DynamiCoS supports end-users to perform automatic discovery, selection and composition using Semantic Web. They have developed prototype for healthcare scenario, which is similar to my prototype model. In this work, authors have not considered non-functional characteristics and keep that as a
future scope. I have incorporated QoS parameters into service discovery, selection and composition process to enhance the user satisfaction.

(Rodriguez Mier et al.) presented a theoretical analysis of graph-based service composition in terms of its dependency with service discovery. Driven by this analysis a composition framework is defined by means of integration with fine-grained I/O service discovery that enables the generation of a graph-based composition, which contains the set of services that are semantically relevant for an input-output request. The proposed framework also includes an optimal composition search algorithm to extract the best composition from the graph minimizing the length and the number of services, and different graph optimizations to improve the scalability of the system.

(Küster et al.) proposed an approach to integrate service composition into service discovery and matchmaking to match service requests. Authors have discussed general issues involved in describing and matching services and present an algorithm implementing the work.

(Ngan and Kanagasabai) proposed a generic framework for Semantic Web Services (SWS) discovery, where discovery and selection are defined as the key tasks of the framework. They have described the benchmarks available to evaluate service discovery system; among these I have used WSC09 ([Kona et al.) datasets. Authors have focused on real life application of SWS discovery systems as an important problem with support of QoS feature. I have worked in the same direction to provide the solution.

(Bellur, Gupta, and Vadodaria) proposed various semantic matchmaking algorithms (i.e. Greedy, Bipartite matching & DL algorithm) based on functional and non-functional requirements of Web service. They have focused on comparative study rather than experimental evaluation of the algorithms.

The work of ([Zeng et al.], (Zeng et al.) focuses on dynamic and quality-driven selection of services by adopting global planning to find the best service components for the composition. They used (mixed) linear programming techniques to find the optimal selection of services. Similar to this approach, ([Ardagna and Pernici), (Ardagna}
and Pernici) extends the linear programming model to include local constraints. Linear programming methods are very effective when the size of the problem is small. Nevertheless, these methods suffer from weak scalability due to the exponential time complexity of the applied search algorithms.

(Laliwala et al.) proposed an approach, which combines workflow, semantic and rules technology to achieve event-driven dynamic composition. This approach uses backward-chain methods for verification of precondition, input and output of services and forward-chain method with ECA rules for generation of composition schema as per BPEL standards.

METEOR-S (Verma et al.) is a Semantic Web-based framework for service composition, which offers mechanisms for semantic description of services, service discovery, and service composition. However, it is a static level composition using a template-based approach of processes. Opposite to this, I have focused on dynamic composition approach.

(Fujii and Suda) presented CoSMoS model for semantic description of services at different levels, i.e., at the data, semantic and logic for dynamic composition. In this approach, focus is given on composition process without considering performance metrics, while I have considered parameters like throughput and response time to measure the performance.

(Kona et al.) proposed an approach to perform discovery and composition of Web services semantically. A multi-step narrowing algorithm is used to perform the composition. Since Prolog with Constraint Logic programming are used to perform the discovery and composition processes, services are pre-processed from USDL (Sheng et al.) and transformed to Prolog terms. Pre-processing process is time consuming, especially for the dynamic service composition. They have not considered non-functional characteristic in the approach.

(Talantikite, Aissani, and Boudjlida) presented a model for automatic Web services discovery and its composition. In order to understandable descriptions, Semantic Annotation is used for Web service discovery and composition. The proposed approach
uses from an inter-connected network of semantic Web services describing in OWL-S using the similarity measure between concepts like pellet before any submitted request. Their proposed approach gives several composition types: serial, dependent parallel and independent parallel. The Semantic Network is explored in backward chaining and depth-first in a single pass. At the end, several composition plans are obtained that satisfy the request and only one optimal composition plan using QoS is returned to the requester.

(Paikari et al.) proposed an automatic framework for composition of Semantic Web services in P2P network. The framework is modeled by Multi-agent System Engineering methodology, which is a famous agent oriented methodology and a top-down approach. This approach is consist four agents: UI Provider, Service Finder, Service Provider and Composer. The composition process is performed through several steps. At each step, composer sends its request for proper next Web service to a service finder.

An approach for QoS based Dynamic Web Service Composition have been proposed by (de Oliveira Jr and de Oliveira), which consists of three parts: Semantics, Syntax and Implementation. Semantics part contains Domain Ontology, Composer, Execution Engine and Service Repository modules. Authors have used a Greedy search approach, which is based on Greedy Algorithm to sort the list of known compositions in descending order through comparator function as a heuristic. Greedy search algorithm is a graph search algorithm, where a service is selected to be expanded based on a heuristic function. My proposed work on SOAP-based Web services is an extension of this work by integrating matchmaking and selection tasks with the service composition problem.

(v.) Other Approaches

(Mallayya, Ramachandran, and Viswanathan) proposed a QoS-based automatic web service composition framework by considering multiple QoS requirements as well as the user preferences. They also proposed a user preference based ranking algorithm, where user can specify his preferences over QoS parameters. This proposed
Chapter 3. Related work

approach composes Web services dynamically and generates the composition plan automatically. Although, authors have proposed automatic composition approach they have not mentioned the use of Semantic Web or AI technique in the proposed work.

(Gholam Hassan Tabatabaei et al.) proposed SCAIMO framework to fulfill security requirements of both service requesters and providers using secure task matchmaker. A prototype called SCAIMO-composer is developed for the validation of proposed work.

(Al-Masri and Mahmoud) proposed a Web service relevancy ranking function (WsRF) based on QoS parameters to find the best available Web services during discovery process based on a set of client QoS preferences. This work is focuses only on syntactic discovery process.

(Michlmayr et al.) proposed the QoS-aware Vienna Runtime Environment for Service-oriented Computing (VRESCO) for dynamic binding, invocation and mediation. Authors describe non-functional attributes in their service meta-data model. These QoS attributes can be specified manually using a management service or measured automatically and integrated into VRESCO environment.

(Vu, Hauswirth, and Aberer) presented a QoS-based Web service selection and ranking algorithm with trust and reputation management support. Authors have given a formal description and validation of the approach with experiments to demonstrate the quality of results under different cheating behaviors. This work mainly concentrates on the QoS-based Web service selection task.

3.1.2 SOAP Web services-based prototypes for Healthcare sector

(Hristoskova et al.) presents a Dynamic Composer for medical services. Based on a semantic description of the medical support services, this Composer enables a service to be executed by creating a composition of medical services. For this purpose, a system is implemented composing the medical services automatically through composition algorithms. The composition is achieved using various algorithms satisfying certain quality of service
(QoS) constraints and requirements. In addition to the automatic composition, the paper also proposes a recovery mechanism in case of unavailable services. When executing the composition of medical services, unavailable services are dynamically replaced by equivalent services or a new composition achieving the same result.

Process oriented semantic healthcare service composition framework using HL7 ontology and Semantic Web services has been proposed by ([Wang et al.]). This approach performs service discovery and composition at design time in a semi-automatic manner which uses semantic profiles to specify the semantic descriptions of process operations using domain specific Ontology, such as HL7. This approach focuses on process oriented composition using WSBPEL which makes it less flexible and inappropriate for the dynamic environment of healthcare, whereas I have developed approach for dynamic and automated service discovery and composition.

([Lee et al.]) proposed Semantic Medical Services (SMS) architecture for service discovery and composition using three types of compositional knowledge. Syntactic, semantic and pragmatic knowledge play a major role in discovery, selection and composition of Web services. They have implemented the framework in a cardiovascular domain. In this framework, authors have not considered user service request as well as performance evaluation.

COCOON is a web services based project aimed at reducing medical errors. This project focuses on resolving the problem of integration in healthcare domain through discussion of the problem of integrating components from service discovery to service composition. It is a WSMO compliant project and uses WSMO compliant service discovery engine for resolving the service discovery issue. In COCOON, the most appropriate services are discovered to be used by the specialist hence providing better healthcare services ([Della Valle and Cerizza]).

Artemis ([Dogac et al.]) is another project based on Semantic Web services for the semantic discovery and composition of services. It uses OWL-S as the approach for implementing semantic web services and uses HL7 as a standard for communication. Artemis uses OWL mapping tool (OWLmt) for the communication between sender and receiver providing semantic interoperability. OWLmt works as a mediator between sender and receiver by comparing sender ontology instances and receiver ontology instances with each other.
for making possible the communication (Della Valle et al.). The primary focus of Artemis project is on data interoperability aspect by resolving heterogeneities between HL7 standards V2 and V3.

3.2 RESTful Web services discovery, selection and composition

This section is divided two subsections. First subsection describes the existing approaches related with my work while second subsection presents the application of Web services-based approaches in healthcare sector.

3.2.1 Approaches for Web service discovery, selection and composition

The characteristics of RESTful Web services are fundamentally different than SOAP-based Web services, so approaches developed for SOAP-based Web service discovery, selection and composition are not suitable. Although the research community has put significant effort on automating SOAP-based Web services, automated RESTful Web service composition problem is less explored (Zhao and Doshi).

An approach presented by (Saquicela, Vilches-Blázquez, and Corcho) to automate the semantic description of RESTful services using a cross-domain ontology, DBpedia and external resources such as synonyms and suggestion services to annotate the parameters of the RESTful services. I adopt similar concept to annotate RESTful services. Another approach, SWEET (Maleshkova, Pedrinaci, and Domingue) facilitates semantic descriptions of Web APIs for developing mashups composed of Linked Open Data and Web APIs.

(John and Rajasree) proposed RESTDoc - that uses a microformat to describe, discover and compose RESTful services by adding semantics. They have defined two kind of discovery :(i) discovery as you browse for HTML link and (ii) Discovery as a service or automated discovery for linking related services, which enables automatic discovery and composition. The solution proposed by authors focuses on a microformat-like syntax to annotate the RESTful services and on an adapter for RDF conversion, whereas in my ap-
Chapter 3. Related work

approach service discovery is achieved from a specified resource by generating the list of associated resources in the form of base URIs and subsequent graph construction mechanism is used to compose the resources.

RESTdesc (Verborgh et al.) represents the semantics of Web services using N3 format, and integrates existing standards and conventions, such as Link headers, HTTP OPTIONS, and URI templates for discovery and interaction. This approach is based on top of RESTful principles and consists of a semantic mark-up model. In the proposed approach, authors have concentrate on description and discovery of services only.

(Pautasso) proposed an extension to BPEL for a composition of REST and SOAP Web services. REST services are wrapped in WSDL descriptions to generate a BPEL composition. My proposed approach highlights a native composition of RESTful services, rather than using technologies of SOAP Web services.

(Bennara, Mrissa, and Amghar) proposed an approach for service composition based on the REST and Linked Data. The proposed solution depends on the description of web resources to support a unified discovery process and composition, where the interactions among resources can be tracked themselves, and publish these traces, through the progressive exploration of resource descriptors. In my approach, I have followed the same principle for discovery and composition with incorporation of additional information on resources such as QoS value for the selection of matched resources.

iServe (Pedrinaci et al.) provides platform for publishing services for discovery and use by exposing service descriptions as Linked Data expressed as a simple vocabulary for services, such as SOAP-based services and RESTful services. In my work, Linked Data principles are used for discovery, selection and composition tasks.

(Zhao and Doshi) explored the RESTful service composition problem using a situation calculus based State Transition System (STS) as a formal model to automate the composition process of RESTful Web services. Authors have identified the challenges of resource description language and data heterogeneity. I have considered LOD based RDF model to describe the RESTful Web services and dereferencing mechanism to compose them.

Algorithms for automatic Web API discovery and composition are proposed by Yong-
Ju Lee et al. (Lee). The discovery algorithm can generate optimal plans by applying strategies that rapidly prune APIs that are unmatched to the query, while the composition algorithm is based on a graph-based approach, where composition candidates are gradually generated by forward and backward searching over the graph. This work considers discovery and composition of Web APIs, such as SOAP and REST using ontology based approach while my proposed approaches cover discovery, selection and composition of RESTful service using Linked Data principles to get the benefit of REST.

Data-Fu (Stadtmüller et al.) enables the development of data driven applications that facilitate the RESTful manipulation of read/write Linked Data resources. For interactions between web resources, this work introduces a lightweight declarative rule language called Data-Fu with state transition systems as formal grounding.

Karma (Knoblock et al.), a semi-automatic approach presented for mapping structured sources to ontologies in order to build semantic descriptions as source models. It also provide a graphical user interface for a user to interactively refine the models, which can be used to convert data into RDF with respect to a given ontology or to define a SPARQL end point to work with an ontology. Demonstration of Karma tool allows users to extract, link, and integrate geospatial data from various sources.

REST-based method is used by Bohara et al. (Bohara, Mishra, and Chaudhary) to solve certain limitations, such as tightly coupled invocation, performance in terms of response data, non-uniform interface and no hyperlink support. In this work, Web Processing Service interface specification provides an approach to publish and execute geo-processes on the Web. Geo-spatial analysis plays critical role to generate alerts and recommendations. Authors have provided android based user interface with local language support but doesn’t provide performance measurement detail of the proposed algorithm.

To minimize the manual efforts and to get the results faster and automated, Alice et al. (Alice, Abirami, and Askarunisa) and (Zhang et al.) proposed the use of Semantic Web Technology like, OWL, RDF and SPARQL to retrieve the information from the heterogeneous documents efficiently. These works propose a model for storing the content of a DOC/HTML document in the RDF format and thereby enabling the information retrieval using SPARQL. Efforts have been done for information retrieval rather than discovery and
composition, which are main objectives of my work.

Page et al. (Page, De Roure, and Martinez) presented an approach about how web service gives semantic results using RDF and linked open data. In this work, author has discussed about domain driven approach without representing framework and algorithm detail.

Mouhoub et al. (Mouhoub, Grigori, and Manouvrier) presented a SPARQL-driven approach for searching linked data and relevant services. An approach for service discovery and composition has been proposed for LOD-based SWS services. Author has not presented implementation and experimental detail of the work.

Hypermedia-Driven RESTful service composition (Alarcon, Wilde, and Bellido) performs dynamically when a client interacts with a server. This approach utilizes a hypermedia-centric REST service description called the Resource Linking Language (ReLL) and Petri Nets as a mechanism for describing the machine-client navigation.

OmniVoke (Li et al.) presented the framework for automated invocation of Web APIs. This work provides a framework that supports the invocation of most Web APIs available over the Web. This framework focuses on semantic description of Web APIs using the extended service model. The presented work focuses on semantic annotation and invocation part, whereas I focus on discovery and composition of Web services.

3.2.2 RESTful Web services-based prototypes for Healthcare sector

Ji et al. (Ji) has shown that how to extract patient-generated or reported social health data, open government health data, and clinical research data from heterogeneous data sources, and then link and combine it to develop an integrated semantic network model. In this work, framework is presented but they haven’t given any detail about proposed approach.

Sheth et al. (Sheth, Anantharam, and Thirunarayan) considered semantic perception algorithms to transform raw data to human intelligible abstractions using medical domain knowledge. A framework called Knowledge-enabled Health(kHealth) is proposed to empower an individual in their health and fitness needs. The key features of kHealth are based on use of (a) relevant personal, public and population level health signals, (b) Semantic Web technologies that include application of relevant medical knowledge in the forms of
ontologies & reasoning and (c) intelligence at the edge by performing all computation on
the users mobile device.

(Parra et al.) proposed a RESTful discovery and eventing specification based ap-
proach, which is implemented in the context of an assisted living healthcare scenario for
connecting heterogeneous sensor and actuator devices in order to obtain real-time health
data from the inhabitants for better health monitoring.

Andry et al. (Andry, Wan, and Nicholson) presented the REST-based architecture for
creating a REST API integrated with a mobile application (iPhone/iPad) that offer physi-
cians, access to their patient’s health records via a community, regional or state Health
Information Exchange (HIE). They describe the architecture of the system which address
security and privacy concerns, the REST API operations and HL7 subset data format used
for lab results and observations.

3.3 Cloud services discovery, service selection and service
composition

This section presents previous studies related to cloud service tasks, such as matchmaking,
selection, and composition. To support the present research, we consider earlier works
related to cloud services, the proposed approaches to cloud service tasks, and the usage of
cloud services in the healthcare sector.

3.3.1 Approaches for Web service discovery, selection and composi-
tion

Joshi et al. (Joshi, Yesha, and Finin) proposed a life cycle model of cloud services with five
phases: requirements, discovery, negotiation, composition, and consumption. The authors
presented an integrated ontology to automate the cloud service lifecycle and provided a
cloud storage service case study to find and achieve storage on the cloud by presenting the
service attributes and policies. Our proposed approach was inspired by this previous work
and others related to the Web service composition life cycle.

Ngan et al. ([Ngan and Kanagasabai) proposed the system supports dynamic semantic
Chapter 3. Related work

matching of Cloud services described with complex constraints and brokering policies currently in use for matching the user need at runtime, as well as the service usage cost. The approaches are based on formal description of service functionality using the SAWSDL, OWL-S and WSMO description languages.

Han et al. (Han and Sim) presented Cloud Service Discovery System (CSDS) for users to perform discovery over the Internet, that consults an ontology when retrieving information about Cloud services. This work focuses on discovery process whereas, I have developed discovery mechanism by incorporating non-functional parameters.

(Sun et al.) carried out a survey of approaches to cloud service selection by considering the seven perspectives related to context, selection and criteria weight. After analysing the reviewed approaches based on these seven perspectives, research issues related to contemporary cloud service selection were identified.

(Wang et al.) proposed QoS-aware service selection approach. Authors have focused on importance of QoS parameters as a part of SLA to select efficient services and assign ranking for service selection. I have used the QoS model provided by (Wang et al.) to propose improved approach for service selection.

(Garg, Versteeg, and Buyya) proposed the Cloud service ranking framework by evaluating Cloud services using Cloud Services Measurement Initiative Consortium (Siegel and Perdue), where Cloud services are ranked using the analytic hierarchy process technique (AHP), which consists of three main steps- problem decomposition, priority judgment and priority aggregation.

(Zhang et al.) proposed a system called CloudRecommender for the automatic selection of infrastructure cloud service configurations. The CloudRecommender supports a selection based on previous stored service information but does not allow a runtime selection using dynamic QoS information.

Some researchers have worked on the Cloud services selection problem with a specific service type, such as SaaS, IaaS, or PaaS. Godse and Mulik (Godse and Mulik) have proposed an AHP-based SaaS service selection method to score and rank services. An AHP hierarchy is prepared to define SaaS service attributes. The relative importance of service attributes is weighted by adding user preferences and domain experts' opinions.
Karim et al. (Karim, Ding, and Miri) presented the techniques for aggregation of multi-level QoS parameters of Cloud services, which supports the ranking and selection of IaaS and SaaS services based on user requirements. In this approach, an AHP hierarchy of a Cloud service weighting model is defined, where QoS criteria of both SaaS and IaaS are considered for the ranking. The proposed AHP hierarchy allows the generation of QoS weights, and the usage of the mapping rules to achieve a best service combination of IaaS and SaaS.

Menzel et al. (Menzel, Schönherr, and Tai) developed a CloudGenius for selection of best combination of a VM image and a cloud infrastructure service to support web server migration to the cloud for IaaS selection. The proposed approach is based on analytic hierarchy process (AHP). The prototype implementation presented is limited to criteria like popularity, price and performance and supports only Amazon-web services.

Jrad et al. (Jrad, Tao, and Streit) proposed architecture for a Cloud service broker for Intercloud environment, where broker aims to find the suitable cloud provider while satisfying the users’ service requirements in terms of functional and non-functional SLA parameters. In this work, authors have proposed matchmaking approach for cloud services. I have extended this approach and proposed improved work for Service discovery using QoS as SLA parameters.

From the perspective of cloud computing, (Sim) contributes to the field of cloud resource management by devising several novel approaches for facilitating cloud service discovery, service negotiation, and service composition. Their work included the development of agent-based search engine, negotiation technique and cooperative problem-solving technique to automate the cloud service composition.

An approach for workflow engine was proposed by (Höing) to integrate different Cloud Computing platforms and services, offering Orchestration as a Service (OaaS). However, the prototype implemented had some missing features, such as automatic service se-
lection, mechanisms for failure/fault handling and adaptability at runtime, use of quality metadata, and SLA support.

Cavalcante et al. (Cavalcante et al.) presented Cloud Integrator for the service composition offered by various Cloud Computing platforms. In Cloud Integrator composition of services is represented using a semantic workflow. This approach focuses on the composition mechanism of the platform. The metadata-based service selection algorithm and its application is described in a case study.

A novel quality-based cloud service composition approach using QoS attributes proposed in (Ye, Bouguettaya, and Zhou). In this work, the tenure process of university department is considered to demonstrate the cloud service composition. This work focuses on syntactic aspects of composition rather than semantic composition.

Gutierrez-Garcia et al. (Gutierrez-Garcia and Sim) proposed an agent-based technique to composing services in multi-cloud environments. Agents are enabled with a semi-recursive contract net protocol and service capability tables to compose as per the user requirements. The single and multiple cloud schemes applicable to dynamic service composition are not provided.

A generic methodology for the representation of Cloud services was proposed by (García and Blanquer), which uses the WS- Agreement specification for capturing and manipulation arbitrary services using SLA fragments. A SLA composition algorithm composes SLA fragments on the fly in response to user request. This work also focuses on syntactic aspects of composition rather than semantic composition.

(Ye et al.) proposed a Cloud service composition framework, that selects the optimal composition based on an end user’s long-term Quality of Service (QoS) requirements. The proposed framework uses a new multivariate QoS analysis to predict the long-term QoS provisions from service providers’ historical QoS data and short-term advertisements represented using Time Series. To select the optimal service composition, the proposed framework uses QoS time series inter-correlations and performs a novel time series group similarity approach on the predicted QoS values.

Ye et al. (Ye, Bouguettaya, and Zhou) presented a novel quality-based cloud service composition approach. The research focuses on the selection of composition plans
based solely on non-functional (QoS) attributes. It proposes a cloud service composition approach to aid end users selecting and composing SaaS providers and IaaS providers in the cloud environment.

Guidi et al. (Guidi, Giallorenzo, and Gabbrielli) presented an APIasaService (API-aaS) layer as a tool to ease the development and deployment of applications based on the API composition. However, the authors focused only on the development and reuse of the API composition and did not carry out experimental studies.

### 3.3.2 Cloud services-based prototypes for Healthcare sector

Kuo and Alex (Kuo) discussed the demand for a cloud computing platform in the healthcare domain, as well as issues regarding its management, technology, security, and legal aspects, along with case studies. They proposed the HC2SP model for the application of such platform in the field of healthcare.

Parekh et al. (Parekh and Saleena) proposed a framework, which is based on main components of healthcare system: patients, doctors, symptoms and diseases. It enables patients to get required health services, such as to book an appointment with the doctor and to get disease information through data mining clustering techniques and mobile application. This approach has focused on statistical analysis of historical data whereas, I have focused on remote treatment facility based on diagnosis resulted from patient’s symptoms and samples using Cloud services for the beneficial to end users.

### 3.4 Summary

The literature review outlines some remarkable observations, such as Web services composition is still interesting research challenge and it is closely related to the service discovery and selection processes to perform the composition efficiently and accurately. Very few efforts have been made to integrate the service discovery, selection and composition for SOAP-based Web services, i.e. [(Yang and Papazoglou), (Da Silva, Pires, and Van Sinderen)] in these approaches, they have not considered non-function requirements which add user satisfaction and quality guarantee to the solution. Heterogeneous Web services provide features, such as dynamic in nature, interoperability, support for Web and hetero-
geneous environment makes it more suitable for healthcare domain.

Through literature study, I have observed that service discovery, selection and composition are interrelated concepts although little attention has been given to integrate them to enhance the performance and efficiency of the solution using RESTful and Cloud services. Moreover, the application of RESTful services in healthcare sector is still very few.

All the aforementioned facts motivates to develop frameworks and solutions for Heterogeneous Web services by integrating service discovery, selection and composition processes. Moreover to demonstrate the applicability of the work, there is a need to develop prototypes by considering available datasets and developing repository of real Web services.