PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
MOHAMED BOUDIAF UNIVERSITY OF M'SILA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

Thesis presented for the award of the Doctoral Degree

in Computer Science

Option: Intelligent Systems and Machine Learning

By

KHELIL Hathem

TITLE

A semantic web services composition model for supporting the interoperability inside an e-government platform

---

Supervised By: Mahmoud BRAHIMI

Defended on: 19/07/2025, before the jury composed of:

Samir AKHROUF	Professor	University of M'Sila	President
Mahmoud BRAHIMI	M.C.A	University of M'sila	Supervisor
Allaoua HEMMAK	Professor	University of M'sila	Examiner
Farid NOUIOUA	Professor	University of B.B.A	Examiner
Tahar MEHENNI	M.C.A	University of M'Sila	Examiner

Academic Year 2024–2025

"I dedicate this small effort to my family, who have always inspired me to push forward for excellence. Their support and unwavering encouragement kept me going. I will always be grateful to my parents, Leila DJEROU and Naceur KHELIL, for being such sympathetic and encouraging people in my life. Many thanks to my brothers Alla Eddine, Sofiane, Hacen, and Hamza, as well as my sisters Sara and Asma. I will always be grateful to them for their unwavering encouragement and support, especially during difficult moments. Finally, I would like to dedicate this work to my dear niece Leila Yafa as well as to my nephews Wael and Kenan. "

KHELIL Hathem

Acknowledgements

First and foremost, I want to thank Allah for helping me along the way and providing me with amazing family, friends, and mentors who helped me finish my thesis.

Without the blessings and assistance of numerous individuals in my life, my thesis would not have been feasible. I am quite appreciative and thankful for their support and assistance. I would like to thank the following people in particular.

I would like to sincerely thank my supervisor, Professor BRAHIMI Mahmoud, for his unwavering encouragement, patience, and positive energy from the first day of my thesis and related research activities. Throughout my research, his advice and insights were invaluable. Thank you, sir.

My cordial thanks extend to my thesis committee members who accepted to evaluate this modest work. Professor Samir AKHROUF, Professor Allaoua HEMMAK, Professor Farid NOUIOUA, and Professor Tahar MEHENNI.

I owe my gratitude to Professor MCHEIK Hamid from UQAC University- Chicoutimi, Québec, Canada. And Professor Mo Adda from the University of Portsmouth, UK. For promoting my internships, for their encouragement and support, for the valuable comments, inspiration and delightful discussions throughout the writing of our research projects and this thesis.

Endless fractals of thanks go to Professor Khadidja Yachba from ESE Oran, Algeria for her generosity, encouragement, and assistance from a distance and for sharing with me her experience and the most important documents related to my research.

I would like to thank all my colleagues in the Department of Computer Science, University Mohamed Boudiaf, M'sila, and all the members of LIAM laboratory, and Big thanks to all the administration's members in the Faculty of Mathematics and Computer Science.

Abstract

Many countries have embraced the concept of e-government by digitizing their public sector entities. This step has been driven by the need to find effective working mechanisms, even under exceptional circumstances, such as the recent COVID-19 pandemic. Like other countries, Algeria has launched several projects to digitize its governmental and administrative entities. However, most projects have been individual, isolated, and uncoordinated, hindering interoperability and information exchange. Since meeting the administrative needs of citizens and various institutions requires the intervention of multiple administrative and governmental entities, the lack of effective interoperability between these entities is one of the most significant shortcomings of current initiatives. In this context, this thesis proposes a solution to improve e-government performance by stimulating interaction and information exchange among its various stakeholders. The proposed solution is citizen-centric and relies primarily on web service composition technology according to non-functional quality of service standards, supported by semantic web technologies and a citizen-specific ontology. This solution supports information exchange and access and improves interoperability and collaboration among government entities. Through a case study of the healthcare system in Algeria, we aimed to demonstrate the benefits of this solution in improving the efficiency, unification, and responsiveness of public services within the framework of unified e-government.

Keywords: Algerian E-Government, Web Service Composition, Ontology, Semantic Web Service, Interoperability, Healthcare System.

ملخص

تبنت العديد من الدول فكرة الحكومة الإلكترونية من خلال رقمنة مصالح قطاعها العام، وقد حفزت هذه الخطوة ضرورة ايجاد آليات عمل فعّالة حتى في ظل الظروف الاستثنائية كما كان الحال مع جائحة كوفيد-19 الاخيرة. وعلى غرار دول أخرى، أطلقت الجزائر العديد من المشاريع لرقمنة هيئاتها الحكومية والإدارية. ومع ذلك، كانت معظم هذه المشاريع فردية ومعزولة وغير منسقة، مما حال دون امكانية تفاعلها وتبادل المعلومات فيما بينها. وبما أن تلبية الاحتياجات الادارية للمواطنين ومختلف المؤسسات غالبًا ما تتطلب تدخل عدة جهات ادارية وحكومية، فإن غياب الترابط والتشغيل البيني الفعال بين هذه الجهات يُعدّ أحد أبرز أوجه القصور في المبادرات الحالية. في هذا السياق، تقترح هذه الأطروحة حلاً لتحسين أداء الحكومة الإلكترونية من خلال تحفيز التفاعل وتبادل المعلومات بين مختلف اطرافها. يتمحور الحل المقترح حول المواطن، ويعتمد بشكل أساسي على تقنية توليف خدمات الويب وفقًا لمعايير جودة الخدمة غير الوظيفية، مدعومة بتقنيات الويب الدلالي وانطولوجيا خاصة بالمواطن. يهدف هذا الحل إلى تحسين تبادل المعلومات والوصول إليها، وتعزيز التفاعل والتعاون بين الجهات الحكومية. ومن خلال دراسة حالة تخص نظام الرعاية الصحية في الجزائر أردنا توضيح مزايا هذا الحل في تحسين كفاءة وتوحيد واستجابة الخدمات العامة في إطار الحكومة الإلكترونية الموحدة.

الكلمات المفتاحية: الحكومة الإلكترونية الجزائرية، تكوين خدمات الويب، الأنطولوجيا، خدمة الويب الدلالية، التشغيل البيني، نظام الرعاية الصحية

Résumé

De nombreux pays ont adopté le concept d'administration électronique en numérisant leurs entités du secteur public. Cette démarche a été motivée par la nécessité de trouver des mécanismes de travail efficaces, même dans des circonstances exceptionnelles, comme la récente pandémie de COVID-19. À l'instar d'autres pays, l'Algérie a lancé plusieurs projets de numérisation de ses entités gouvernementales et administratives. Cependant, la plupart de ces projets sont restés individuels, isolés et non coordonnés, ce qui a entravé l'interaction et l'échange d'informations. La satisfaction des besoins administratifs des citoyens et des diverses institutions nécessitant souvent l'intervention de multiples entités administratives et gouvernementales, le manque d'interopérabilité efficace entre ces entités constitue l'une des principales lacunes des initiatives actuelles. Dans ce contexte, cette thèse propose une solution pour améliorer les performances de l'administration électronique en stimulant l'interaction et l'échange d'informations entre ses différentes parties prenantes. La solution proposée est centrée sur le citoyen et repose principalement sur une technologie de composition de services web selon des normes de qualité de service non fonctionnelles, soutenue par des technologies du web sémantique et une ontologie spécifique au citoyen. Cette solution vise à soutenir l'échange et l'accès à l'information, et à améliorer l'interopérabilité et la collaboration entre les entités gouvernementales. A travers une étude de cas du système de santé en Algérie, nous avons tenté de montrer les bénéfices de cette solution pour améliorer l'efficacité, l'unification et la réactivité des services publics dans le cadre d'un e-gouvernement unifié.

Mots-clés: Gouvernement électronique algérien, Composition de services Web, Ontologie, Service Web sémantique, Interopérabilité, Système de santé.

Contents

List of Figures

List of Tables

Abbreviations

ADT	Admission, Discharge, Transfer
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
AS	Abstract Service
BDI	Belief-Desire-Intention (Agent Model)
BPMN	Business Process Modeling Notation
BPEL	Business Process Execution Language
BTLBO	Balanced Teaching-Learning-Based Optimization
CCD	Continuity of Care Document
CCR	Continuity of Care Record
CDA	Clinical Document Architecture
CEN	European Committee for Standardization
CRM	Customer Relationship Management
CS	Concrete Service
DBTLBO	Discrete Balanced Teaching-Learning-Based Optimization
DICOM	Digital Imaging and Communications in Medicine
DIP	Data, Information, and Process Integration (EU Project)
EHR	Electronic Health Record
EGDI	E-Government Development Index
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
FHIR	Fast Healthcare Interoperability Resources
GA	Genetic Algorithm
G2B	Government-to-Business

G2C	Government-to-Citizen
G2E	Government-to-Employee
G2G	Government-to-Government
GovML	Government Markup Language
GUI	Graphical User Interface
HCI	Human Capital Index
HL7	Health Level Seven
HIS	Hospital Information System
HQMF	Health Quality Measure Format
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTN	Hierarchical Task Network
HSA	Harmony Search Algorithm
ICD	International Classification of Diseases
ICT	Information and Communication Technologies
IaaS	Infrastructure as a Service
ISO	International Organization for Standardization
IoT	Internet of Things
JMS	Java Message Service
JSON	JavaScript Object Notation
KAON	Karlsruhe Ontology Framework
KIF	Knowledge Interchange Format
LOINC	Logical Observation Identifiers Names and Codes
MAMS	Multi-Agent Microservices
MAS	Multi-Agent System
MDP	Markov Decision Process
MQTT	Message Queuing Telemetry Transport
NEGS	National European e-Government Services
NIC	Nature-Inspired Computing
NSGA	Non-dominated Sorting Genetic Algorithm
OIL	Ontology Inference Layer
OCML	Operational Conceptual Modeling Language
OECD	Organisation for Economic Co-operation and Development

OSI	Online Services Index
OWL	Web Ontology Language
OWL-S	OWL for Services
PaaS	Platform as a Service
PDDL	Planning Domain Definition Language
PEGS	Pan-European e-Government Services
PSO	Particle Swarm Optimization
QoS	Quality of Service
QWSC	QoS-Aware Web Service Composition
RDF	Resource Description Framework
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RPC	Remote Procedure Call
REST	Representational State Transfer
SAWSDL	Semantic Annotations for WSDL and XML Schema
SaaS	Software as a Service
SHOE	Simple HTML Ontology Extensions
SMTP	Simple Mail Transfer Protocol
SNOMED CT	Systematized Nomenclature of Medicine – Clinical Terms
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SSA	Social Spider Algorithm
SWS	Semantic Web Services
TLBO	Teaching-Learning-Based Optimization
TII	Telecommunications Infrastructure Index
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modeling Language
UNDESA	United Nations Department of Economic and Social Affairs
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VMR	Virtual Medical Record
W3C	World Wide Web Consortium

WSC	Web Service Composition
WSDL	Web Services Description Language
WS	Web Service
WSS	Web Service Selection
WSMO	Web Service Modeling Ontology
XML	Extensible Markup Language

General Introduction

The terms ”e-government” and ”e-governance” are often used interchangeably in academic literature and official documents [1]. E-government primarily focuses on the use of digital technologies to improve the effectiveness and efficiency of government operations, including public service delivery, administrative management, and coordination among government agencies. It aims to streamline processes, improve service delivery, and facilitate access to government services through electronic platforms and automated systems. E-governance, on the other hand, goes beyond service delivery to encompass broader aspects of governance, such as citizen engagement, transparency, accountability, and participatory decision-making. Although their objectives differ, e-government and e-governance are fundamentally interconnected and together contribute to shaping the digital transformation of government systems. Both describe how government institutions leverage information technologies, such as the internet, mobile computing, and wide-area networks, to enhance their interactions with citizens, businesses, and other government agencies.

The evolution of e-government has been closely tied to advancements in Information and Communication Technologies (ICT), progressing through several phases. The initial phase relied on static web technologies like HTML and HTTP for basic information dissemination, offering limited interactive services. The second generation introduced dynamic web technologies, including web services, to automate administrative processes and facilitate information exchange, though often lacking user-centric approaches. The third phase focused on integrating technologies to enhance data exchange and unify public services, yet faced significant data standardization and interoperability challenges. The most recent phase has involved adopting Semantic Web technologies, providing standardized descriptions and structures for services and data, thus addressing earlier limitations and advancing the delivery of e-government services.

Since the emergence of e-government in the late 1990s, countries worldwide have pursued various initiatives to harness its potential. The COVID-19 pandemic, however, served as a significant catalyst, significantly accelerating the global adoption of e-government solutions [2]. In the wake of the pandemic, many governments increasingly relied on digital platforms to maintain the continuity of essential public services while minimizing physical interactions. As a result, the transition to online access to healthcare, social protection programs, education, and other services became essential.

The Algerian government has recognized the importance of e-government and has worked tirelessly to implement it at the regional and national levels [3]. According to Idoughi [4], it has prepared and developed numerous plans and strategies as the main pillars for its implementation.

Following the COVID-19 pandemic, Algeria strengthened its commitment to digital transformation, recognizing the crucial role of digital systems in improving interactions between the state and citizens [5]. The pandemic underscored the importance of having a robust digital infrastructure to ensure efficient service delivery during crises.

Despite the progress made in digitizing public services, the Algerian government has resulted in government agencies and public administrations providing separate and independent e-services without considering user needs. The absence of a central system connecting various departments has led to fragmented services, with each institution managing its digital platform, often using different technologies and standards. This lack of integration leads to inefficient data exchange, inconsistent service delivery, and a complex user experience, as citizens must navigate multiple systems. Creating a user-centric, interoperable platform that combines services across institutions is critical to streamlining access, improving service delivery, and promoting a more efficient, transparent, and accessible e-government.

To address interoperability and integration challenges, recent advances in semantic web technologies [6] offer a robust platform for a knowledge-based, user-centric, distributed, and interoperable e-government. By enhancing data and service descriptions, semantic technologies provide a structured way to represent and exchange information, achieved through ontologies and Semantic Web Services (SWS). Ontologies serve as formalized domain knowledge representations, fostering shared understanding and seamless data sharing across governmental entities [7, 8]. On the other hand, Semantic Web Services describe intelligent services, enabling automated processes like service discovery and orchestration. Vitvar et al. [9] highlight two research strands: one focuses on e-government as a testbed for evaluating SWS maturity, while the other explores SWS as a foundation for interoperable government services.

A closely related concept is Web service composition, which involves integrating multiple individual services into a unified, coherent application [10]. This integration enhances the flexibility and scalability of e-government systems, enabling the dynamic assembly of more complex services from simpler ones. Web service composition thus plays a critical role in achieving seamless interoperability and enhancing the overall efficiency of government service delivery.

The e-government domain is inherently complex, involving many concepts, services, business rules, and stakeholders, including public administrations and citizens. In Algeria, this complexity is further exacerbated by fragmented public information systems and administrative processes, resulting in a lack of unified knowledge representation. This fragmentation hinders efficient service delivery and integration across governmental institutions. Therefore, the primary goal of Algerian e-government should be to establish a secure, unified platform capable of delivering consistent, high-quality services while integrating multiple governmental functions.

Motivation and Research Problem

This thesis is motivated by addressing current issues of e-government systems. It examines how the potential of Semantic Web technologies can facilitate the development of a more user-focused, interoperable, cohesive digital government. The research concentrates on how Semantic Web technologies can bridge the disjoints of government service delivery via a user-centred approach that will provide better data sharing, service discovery, and harmonization of processes. Special attention is given to ontologies and Web service composition as key enablers of integration and improved service delivery. Hence, the main research question of this thesis discusses the following:

”How can Semantic Web technologies support the development of a unified, interoperable e-government platform centred on the user in Algeria, enabling easy integration among different government institutions?”

Consequently, the following specific research questions are suggested to answer this overall question:

- How can the Algerian e-government platform utilize ontology to standardize knowledge representation and enhance interoperability among government entities?
- How can semantic web service compositions promote the interoperability of an e-government system while enhancing integration and collaboration across governmental institutions?
- How can web service composition automate and simplify complex government processes, fostering efficiency and service delivery in Algeria?

By addressing these questions, the research aims to contribute to the development of an integrated, efficient, and citizen-oriented e-government system in Algeria.

Contributions of the Thesis

The main contributions of our thesis can be summarized in three points:

First, we propose designing a comprehensive e-government framework based on ontologies and web service composition tailored to the Algerian context. This framework aims to enhance interoperability, optimize administrative processes, and deliver more efficient, user-focused services by addressing fragmentation within public sector systems.

Second, we introduce methods for optimizing web service composition to improve interoperability and cost-efficiency within Algerian e-government infrastructures. The study focuses on intelligent service selection and composition strategies that enhance government platforms' quality, responsiveness, and agility while reducing development time and associated costs.

Third, we present a case study on the Algerian healthcare sector to demonstrate the practical application of the proposed approaches. This case study shows how ontologies and semantic web service composition can improve the secure exchange of medical records, strengthen institutional collaboration, and support more integrated and efficient patient care. The findings highlight the broader applicability of the proposed models beyond the healthcare sector to other areas of public administration.

Organization of the Thesis

We have structured this thesis into five chapters to achieve our goals, each addressing specific aspects of the research and progressively leading to the proposed solutions.

The first chapter provides an overview of e-government systems, examines the current state of e-government projects in Algeria, and highlights achievements and ongoing challenges.

The second chapter explores web services concepts, structures, uses, benefits, and issues.

The third chapter presents semantic technologies in web services, especially their composition, applications, tools, and the possible advantages of merging semantics with service composition to address e-government issues.

The fourth chapter offers the key research contributions of the conceptual model for e-government interoperability, the creation of a citizen profile ontology, and the optimization strategy for web service composition depending on an improved Balanced Teaching-Learning-Based Optimization (BTLBO) algorithm.

In the fifth chapter, and using our contributions, we present a case study concerning the Algerian healthcare system to show our proposed model's usefulness.

Finally, we conclude our thesis with a conclusion and some perspectives.

Chapter 1

E-Government

1.1 Introduction

E-Government refers to integrating Information and Communication Technologies (ICTs) to enhance and transform how governments deliver services to citizens, businesses, and other entities. E-government utilizes the Internet, mobile computing, and wide area networks to optimize administrative operations, improve service delivery, increase transparency, and promote public involvement. E-government is the digitization of government services; it also encourages the empowerment of people and companies through better access to information and enhanced communication tools.

The chapter discusses the evolution and context of e-government, including a description of the key components needed for successful e-government. We provide an overview of the different types of e-government applications and their contributions to public sector service delivery and activities. The chapter also takes a critical look at the current situation around e-government in Algeria, including investigating the successes and shortcomings in developing and implementing government e-services. Finally, the chapter discusses future directions and possible areas for development and research in e-government.

1.2 Definitions

E-government refers to the use of ICT by government agencies to transform and enhance their operations, both internally and externally, with a focus on improving interactions with citizens, businesses, and other government entities. Over time, the concept of e-government has evolved and various definitions have emerged.

According to the World Bank [11], e-government involves leveraging ICT to transform government interactions with stakeholders, while the Organization for Economic Cooperation and Development (OECD) [12] defines it as a means to improve the management of public affairs through technology. The United Nations Department of Economic and Social Affairs (UNDESA) [14] emphasizes the role of ICT in both enhancing internal government operations and fostering external relationships.

E-government is defined in [15] as the use of digital tools and institutional frameworks to improve government responsiveness, advance social justice, and bridge the digital divide by enabling efficient, integrated, and citizen-centric public service delivery through online and mobile platforms.

It encompasses four key dimensions: e-services, which deliver government services via the internet; e-management, which enhances government operations and internal management; e-democracy, which promotes greater citizen participation in the decision-making process; and e-commerce, which facilitates online transactions for goods and services. These dimensions work together to improve efficiency, accessibility, and engagement between the government and its citizens.

Other definitions, such as those by Bhatnagar [16], highlight e-government's role in enhancing administrative efficiency, reducing corruption, and promoting transparency.

From a citizen-centric perspective, e-government is defined by Gichoya [17], as a means to enable citizens to access government services and information online, anytime and from any location. This approach emphasizes enhancing convenience and accessibility for the public, allowing them to engage with government services more efficiently and effectively.

Vephkhvia in [1], differentiates between e-government and e-governance as two separate concepts. E-governance is a broader term that covers a wide range of relationships and networks within the government, focusing on the use and application of ICT to enhance various governmental processes. In contrast, e-government is a more specific concept, primarily centered on the creation of direct online services for citizens, with a particular focus on services such as e-taxes, e-education, and e-health.

Government involves the use of information and communication technology (ICT) by public authorities to foster active and direct participation of citizens in the decision-making process [18], In this context, e-government serves as a tool to enhance citizen engagement in a variety of governmental activities, ranging from policy formulation to the execution of government programs and projects.

E-government allows the government to consolidate various information systems and business processes into a unified digital platform, thereby enhancing and speeding up public access to government services. This integration encompasses a wide array of services, including public administration, healthcare, education, business licensing, tax payments, and beyond. With the use of electronic systems, processes such as registration, document submission, and payments can be handled online, effectively reducing bureaucracy and minimizing waiting times[19].

1.3 Historical Evolution

The term "e-government" emerged in the late 1990s, although the use of computers in government agencies dates back to the early stages of the technological revolution. Government initiatives, societal demands, and technological advances have influenced the historical development of e-government [20, 21]. E-government has evolved over time in several phases, each of which has improved the security, efficiency, and personalization of interactions between citizens and government by building on the previous phase [2, 3, 20]. Different developments in service delivery and technology integration define each phase. An overview of these stages is provided in Table 1.1, along with information on the technology used, the purposes of e-government, and the types of interactions that take place between government and its citizens.

The development of e-government is progressing at different rates across regions. The United Nations e-Government Survey tracks this progress and provides accurate, data-driven analyses at the national and regional levels [13]. It assesses results using the

Stages	**Purpose**	**Government-Citizen Interaction Nature**	**Technology Used**
Before the 1990s: Pre-Internet Era	Introduction of computers in government to improve the efficiency of bureaucratic processes such as data storage and management.	No direct interaction between citizens and government services via technology.	Basic computers, data storage, and management systems.
1990s: Internet Adoption	Establishing websites to disseminate public information, including government resources, laws, and regulations. Make government information more accessible and inform citizens of important updates.	One-way communication from government to citizens.	Internet and early website development.
Late 1990s to Early 2000s: Transactional Services	The focus shifted from disseminating information to facilitating direct communication between citizens and government through digital means.	Beginning of two-way communication.	Online platforms, secure payment systems, and digital forms.
Mid 2000s: Multipurpose Portals	Enhancing citizen engagement, personalizing services, and improving the overall democratic process by making services and decision-making more interactive.	Creating multi-functional portals that provide users with a single point of entry for all government services.	Advanced internet platforms, online feedback systems, and interactive tools.
2010s and Beyond: Transformational E-Government	Establishing a more open, transparent, and effective government to boost innovation, decision-making, and service delivery.	Creating a more efficient and citizen-centric government through data-driven decision making, open data, and digital platforms.	Cloud computing, big data, artificial intelligence (AI), and mobile technologies.

Table 1.1: Stages of Historical Development of E-Government

E-Government Development Index (EGDI), a composite indicator derived from the weighted average of three standardized sub-indices: the Telecommunications Infrastructure Index (TII), the Human Capital Index (HCI), and the Electronic Services Index (ESI):

- Online Services index (OSI): Analyzes the quality and accessibility of digital platforms and electronic services for citizens.
- Telecommunications Infrastructure Index(TII): Assesses access to the Internet and communications technologies.
- Human Capital Index(HCI): Measures the education levels and digital skills of the population.

Member States are classified into four main groups: very high, high, medium, and low, based on the indicator values. The survey assesses progress in e-government development at the global, regional, national, and local levels, providing insight into the key digital principles needed to achieve the Sustainable Development Goals. The United Nations e-Government Surveys conducted between 2008 and 2022 are presented in Figure 1.1.

The range of EGDI values for each group is mathematically defined as follows: very high EGDI values range from 0.75 to 1.00; high EGDI values range from 0.50 to 0.7499; medium EGDI values range from 0.25 to 0.4999; and low values range from 0.0 to 0.2499.

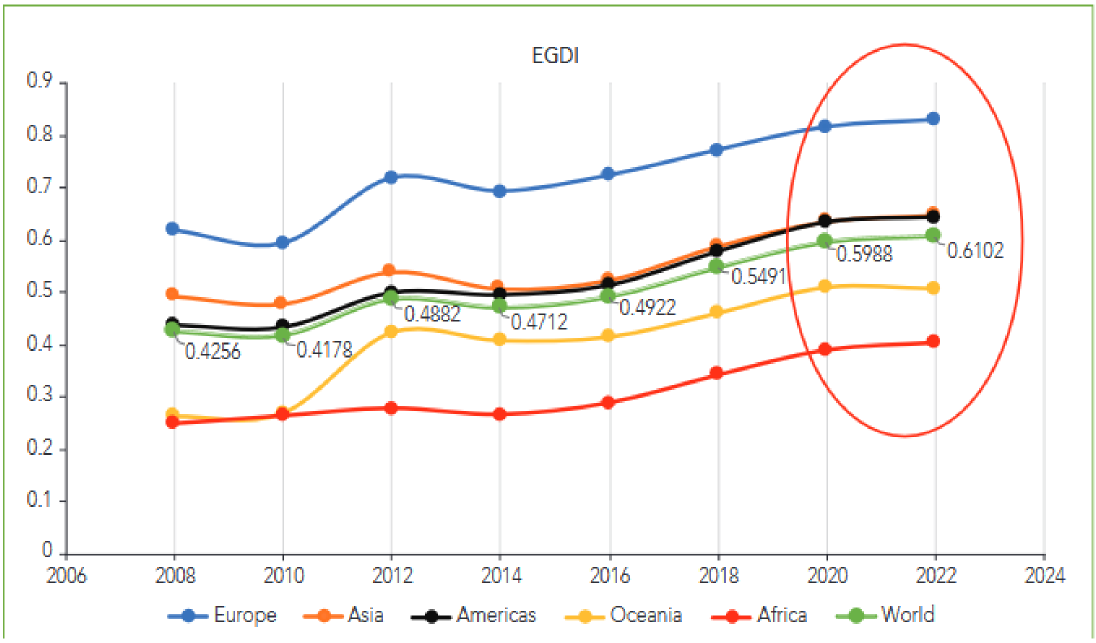

FIGURE 1.1: EGDI Global and Regional Average Value [13]

As shown in Figure 1.1, the overall e-Government Development Index (EGDI) increased overall due to the COVID-19 pandemic, which had a significant impact on e-government development globally. From 2020 to 2022, most countries focused on managing the pandemic, prioritizing the delivery of online services in health, education, social protection, and, in some cases, justice [2]. The most significant expansion of online services was observed in the social protection sector. The number of countries offering national portals allowing users to apply for benefits such as maternity care, child benefits, pensions, housing, and food assistance has increased by 17 % since 2020 [13].

1.4 Importance of E-Government

E-governance offers a wide range of benefits to governments and citizens, going far beyond the simple application of ICT infrastructure to information management.

1.4.1 Improving Service Quality

Improving the quality of services has been recorded as the most important benefit associated with e-governance by several other authors [20, 22, 23, 25]. E-government service quality can be assessed through five key dimensions [24]: ease of interaction, fulfillment, citizen care, security and privacy, and trustworthiness. Ease of interaction refers to how effortlessly citizens can engage with government websites or portals. This includes aspects such as user-friendly website design, advanced technology, intuitive software, well-organized information, easy information retrieval, and the smooth downloading of forms. Fulfillment measures the degree to which e-government services align with citizens' expectations and needs, ensuring that public services are readily available and effectively delivered. Security and Privacy pertain to how safe citizens feel when sharing personal or financial information on government websites or portals. It reflects the assurance that such data is protected, not shared with third parties, and used solely for its intended purposes. Citizen Care represents the responsiveness and effectiveness of government customer service agents in assisting citizens. This includes their ability to promptly resolve issues and provide efficient and compassionate support to address citizens' concerns. Finally, Trustworthiness reflects the reliability of government websites in fostering trust among citizens. It encompasses the government's ability to maintain safety, transparency, and integrity in its online interactions and services.

1.4.2 Structuring Government Information

One of the key advantages is the ability to structure government information, which is often scattered across various departments, into a customer-centered format, enhancing accessibility and usability [26]. According to Widiamoko [22], E-government, allows the government to combine various business processes and information systems into a single digital platform, which speeds up and simplifies public access to government services.

This includes a broad range of services, including healthcare, education, business licensing, public administration, tax payments, and more. Using an electronic system reduces bureaucracy and lengthy wait times by allowing the registration process, document submission, and payment to be completed online.

1.4.3 Enhancing the Efficiency of Public Administration

E-governance has the potential to revolutionize government processes, policies, regulations, and administrative procedures, and significantly improve the efficiency of public administration [23, 27]. In traditional systems, delays and errors in document processing often arise due to physical constraints such as distance and time. However, with the implementation of e-government, administrative tasks can be performed electronically, ensuring speed and accuracy. For example, processes such as resident registration, business licensing, and tax payment can be handled online, reducing time and costs for both citizens and companies.

1.4.4 Improving Transparency and Building Trust

In traditional systems, public information is often difficult to access or hidden. In contrast, e-government increases transparency, reduces costs, and allows citizens to track how the government allocates public funds and administers its programs. This openness helps prevent corruption and abuse of power, which builds trust between the government and the public. According to Bertot et al. [28], e-governance serves as an important mechanism for improving individual accountability among government officials.

1.4.5 Facilitating Public Participation in Decision-making

E-government can enhance public participation in decision-making processes. By using electronic platforms, the government can collect input and feedback from citizens on the policies and programs being implemented. This creates opportunities for individuals to participate and influence the development of public policy actively. In this way, e-government supports the strengthening of democracy and fosters closer relations between the government and society [22].

1.4.6 Promoting Economic Stability and Reducing Poverty

E-government has the potential to drive economic development and attract investment. By offering an online platform for business and investment, the government can draw more investments into the digital sector and broaden market access for local businesses. It also streamlines the licensing process and reduces administrative hurdles, which can stimulate economic growth and generate new employment opportunities. Furthermore, e-government fosters collaboration between the government, private sector, and civil society, enabling the development of innovative solutions to social and economic challenges [2].

1.5 Components of E-Government

The key elements that collaborate to effectively deploy and operate e-government applications, improving service delivery, accessibility, and government transparency, typically include the following:

1.5.1 Infrastructure

A strong and effective technology infrastructure is crucial for the successful implementation of an e-government system and the delivery of services [29]. Information Technology (IT) infrastructure encompasses all components of IT, including hardware, software, networks, communication systems, software applications, legacy systems, current organizational technology, and electronic systems [30, 31]. Moreover, the design and integration of various infrastructure capabilities, and their strategic use in a coordinated approach, are essential for the successful deployment of e-government [32].

Layne et al. [33] highlighted the importance of an efficient communications network and infrastructure as a foundational element for integrating information systems (IS) in government institutions. Similarly, Chango [34] emphasized that assessing a country's progress toward full e-government should begin with a clear understanding of its existing infrastructure and policies. According to [35, 36], many governments have identified the lack of technical infrastructure as the primary barrier to the growth of e-government.

1.5.2 Information Technology Services Delivery

In e-government, managing distributed systems and ensuring seamless service delivery are essential. The implementation of advanced IT architectures, such as Service-Oriented Architecture (SOA) and Cloud Computing, effectively addresses these challenges. SOA promotes the integration and interoperability of diverse systems by providing a flexible, modular framework where both existing and new services can interact efficiently, fostering collaboration between various government agencies [37, 38]. Cloud Computing further enhances e-government by offering scalable, cost-effective infrastructure, enabling governments to deploy services with improved accessibility, performance, and reduced reliance on local resources [39]. Together, these architectures provide a robust software framework that supports the seamless interaction of distributed systems in e-government, improving service delivery and fostering better citizen engagement [40].

1.5.3 Data Management Infrastructure

Data management plays a critical role in e-government by supporting the efficient collection, organization, storage, and dissemination of vital information for public service delivery [41]. A strong database infrastructure is essential for securely managing large volumes of both structured and unstructured data, ensuring fast and reliable real-time access and query performance. Interoperability frameworks within the database structure enable seamless exchange and integration of information across various government departments, promoting consistency and minimizing redundancies in operations [42]. With the rise of big data technologies, the database structure is now capable of handling vast and complex datasets, allowing for advanced analytics that provide valuable insights to improve decision-making and enhance the citizen experience [43]. Cloud-based platforms are increasingly utilized to scale storage capabilities, enabling data-driven innovation while maintaining the security and integrity of the system. Effective governance and adherence to privacy regulations are key to ensuring data integrity and maintaining public trust in e-government systems.

1.5.4 Security and Privacy Infrastructure for E-gouvernment

Security and privacy are essential elements of e-government and play a vital role in building trust in digital government services [44]. These elements ensure the protection of citizens' privacy and sensitive information, which is essential for the success of e-government initiatives [45]. A set of strategies that contribute to creating a secure and reliable e-government environment, are required and include [46]: authentication and authorization systems that verify users' identities and grant them access to appropriate services and data. Data encryption is another essential safeguard, protecting sensitive information from unauthorized access during transmission and storage[45]. In addition, strong cybersecurity measures are essential to defend against cyber threats and ensure the integrity, confidentiality, and availability of government systems and services [47].

1.6 Types of E-government Applications

The government offers services to various customers, including citizens, businesses, and government employees. Each service differs based on the specific needs of the customer. The World Bank categorizes e-government into three classes which are expressed through the interactions: government to government (G2G), government and citizens (G2C), and interaction between government and business enterprises (G2B). Researchers [48, 49] have introduced a fourth category, government-to-employee (G2E), expanding upon the World Bank's classification.

- Government-to-Citizen (G2C) interaction includes initiatives designed to enhance citizens' engagement with the government both as recipients of public services and as active participants in decision-making processes. This interaction covers the delivery of public services and opportunities for public consultation and participation. G2C aims to give citizens a portal that makes it simple for them to access the information and services they require. Furthermore, G2C enables citizens to participate in democratic activities and decision-making processes [50].
- Government-to-Government (G2G) Data sharing and electronic communication between government entities are key components of "government to government"

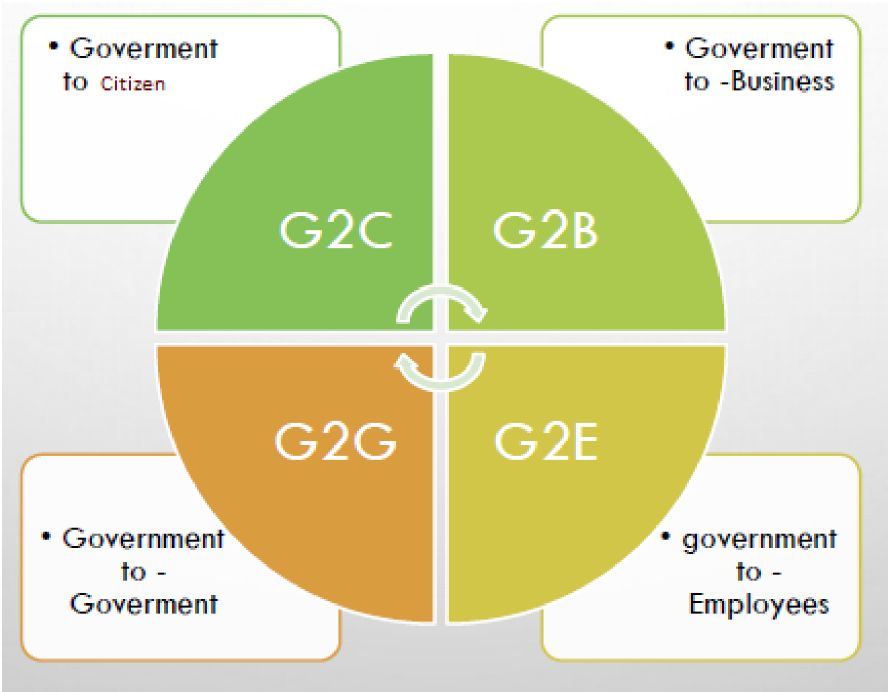

FIGURE 1.2: Types of E-government applications

(G2G) interaction. At the national level, this includes exchanges between institutions as well as between national, provincial, and local levels. Ebrahim in [51] asserts that ”no structuring can exist without sharing, and no organizational intelligence will be created without structuring,” emphasizing that this form of e-government should be implemented before the others. For e-government to be successful, it is crucial to integrate all government departments and agencies to facilitate information exchange.

- Government-to-Business (G2B) interaction involves transactions specific to businesses (such as payments for the sale and purchase of goods and services) and the online delivery of services to businesses. According to Forman [52], G2B is a service that facilitates communication between the government and commercial entities. This connection is crucial for both parties: for the government, it helps reduce redundancy within commercial agencies, and for businesses, it streamlines service processing, saving time and costs, which can positively affect revenue. G2B services include accessing application forms, obtaining new licenses, and renewing existing licenses, among others.

- Government-to-Employee (G2E) provides employee-specific services that aid in their understanding of governmental rules and laws. Furthermore, these services can decrease the number of workers required for a given process, increase administrative efficiency, and shorten the time needed to finish particular tasks [51].

1.7 Situation of E-government in Algeria

Since the emergence of the concept of e-government countries around the world have begun to develop their own e-government initiatives to benefit from the promised benefits. In the 1990s, the United Nations Department of Economic and Social Affairs (UNDESA) [14] initiated e-government projects in many developed countries. E-governance has evolved in phases, starting with the emphasis on computerizing and automating government functions (e-government) and moving toward a more thorough integration of ICT for participatory governance, public engagement, and cooperative decision-making (e-democracy and e-participation) [28].

However, Arab countries began to adopt e-government in the early 21st century. Countries such as Syria, Egypt, Jordan and the United Arab Emirates, which have witnessed the benefits of e-government, have encouraged other Arab countries to implement similar initiatives. Recognizing the benefits, Algeria also sought to implement e-government to leverage the positive externalities generated by UNDESA [3]. In the early 2000s, initiated efforts to digitize and streamline governmental processes. This was accomplished through the creation of a network of ministries and government organizations, each playing a vital role in the development and execution of e-governance initiatives. Key agencies and organizations involved in this effort include [54]:

- **The National Agency for the Promotion and Rationalization of the Use of Information and Communication Technology** (ANPT: Agence Nationale pour la Promotion et la rationalisation de l'utilisation des Technologies de l'information et de la communication) is the main organization responsible for coordinating ICT projects in Algeria. It plays a pivotal role in advancing e-governance initiatives, overseeing the development of e-infrastructure, and promoting the effective use of ICT in the public sector
- **The Ministry of Posts, Telecommunications, Technology, and Digitalization** is tasked with formulating ICT policies, regulations, and strategies to

facilitate the integration of digital technology into government operations and services.

- **The Ministry of the Interior, Local Government, and Spatial Planning** is in charge of putting e-governance programs into action locally. with the goal of enhancing the efficiency and transparency of local government activities.
- **Ministry of Finance:** As the authority responsible for financial management and revenue collection, the Ministry of Finance has been actively involved in the development of online payment systems and the digitalization of financial operations within the government.

The adoption of e-government in Algeria has been influenced by various factors [3, 55, 56, 62], both internal and external. Key factors include the government's commitment to modernization and the desire to improve public service delivery, which have been at the heart of the e-government initiative. Algeria's socio-political environment, characterized by a need for greater transparency and efficiency in public administration, has also played a significant role in the adoption of digital government services. Technological infrastructure, such as expanding internet access and developing digital literacy among citizens and government employees, has been crucial in enabling the transition to e-government. In addition, international influences, such as recommendations from organizations like the United Nations and the European Union, have pushed Algeria to align with global trends in digital governance. However, challenges such as inadequate technological infrastructure, resistance to change among civil servants, and concerns about data security have hampered the full realization of e-government.

Despite these challenges, the Algerian government has continued to advance initiatives such as the e-Algeria project which ran from 2008 to 2013. In this project, the Algerian action plan is organized around thirteen main axes, with a portfolio of work created for each axis and a list of specific key objectives to be achieved by 2013. The description of these axes are presented in [4]. This ambitious and comprehensive initiative aimed to establish an information society in Algeria and bring the administration closer to citizens by offering electronic public services. Nevertheless, the project was criticized for its dependence on public funds, its high expenses because of its ambitious nature, and the challenge of finishing it in the allotted time [57]. Notwithstanding its difficulties, the initiative produced a number of advantages, such as the ”Osratic” project, Internet access for six million Algerians, the creation of technology parks, the introduction of mobile phones, Algeria Post withdrawal cards, the ”Chifaa” card, and more.

To enhance the role of ICT in the country, the government has collaborated with various international organizations. Algeria has introduced numerous e-governance initiatives and services that have significantly transformed the way the government interacts with its citizens. These include [58]:

- Services such as driving license renewals, business registration, and the issuance of birth and marriage certificates.
- The implementation of biometric identity cards, which facilitates secure online interactions with government services and improves security measures. This initiative has simplified the process of identity verification for various transactions.
- The introduction of secure online payment methods, like the El Dahabya card, has made it easier for citizens to pay for government services and fees.
- The digitalization of public administration, encompassing the implementation of electronic document management systems, online public procurement, and the automated delivery of public services.
- Participatory platforms: The government has adopted digital platforms to gather public opinion and involve citizens in the policy-making process, fostering greater participatory governance.

In June 2020, Algeria established a Ministry of Digitization to support the transition to a digital state. According to official documents, the ministry’s role is mainly to gather information and oversee the implementation of digitization projects by other ministries, which already had structures in place for digitization. However, it does not directly influence the planning of these projects. Each ministry sets its own digital initiatives based on its priorities.

The Ministry of Digitization’s 2021 evaluation revealed several issues in the organization and execution of digitization projects [3]:

- Heterogeneity of structures: The entities responsible for digitization within the ministries vary considerably in terms of organization, ranging from a general directorate to a simple office attached to a support directorate.
- Inadequate skills: The qualifications of the people in charge of these entities are often disparate and sometimes inadequate for the functions they occupy.
- Insufficient funding: The lack of budgets specifically dedicated to digitalization in ministries is a major obstacle. Projects are generally financed from credits included in operating or capital budgets,

Despite these efforts, Algeria remains at an intermediate stage of e-government development in global rankings. According to the 2022 survey on country performance in e-government based on the Electronic Government Development Index (EGDI), Algeria ranks 9th in Africa with an EGDI higher than the African average but lower than the global average [53] (see figure1.3). The survey also gives us the values of the three dimensions (TII, HCI, ESI) that were used to calculate Algeria's EGDI (figure 1.4):

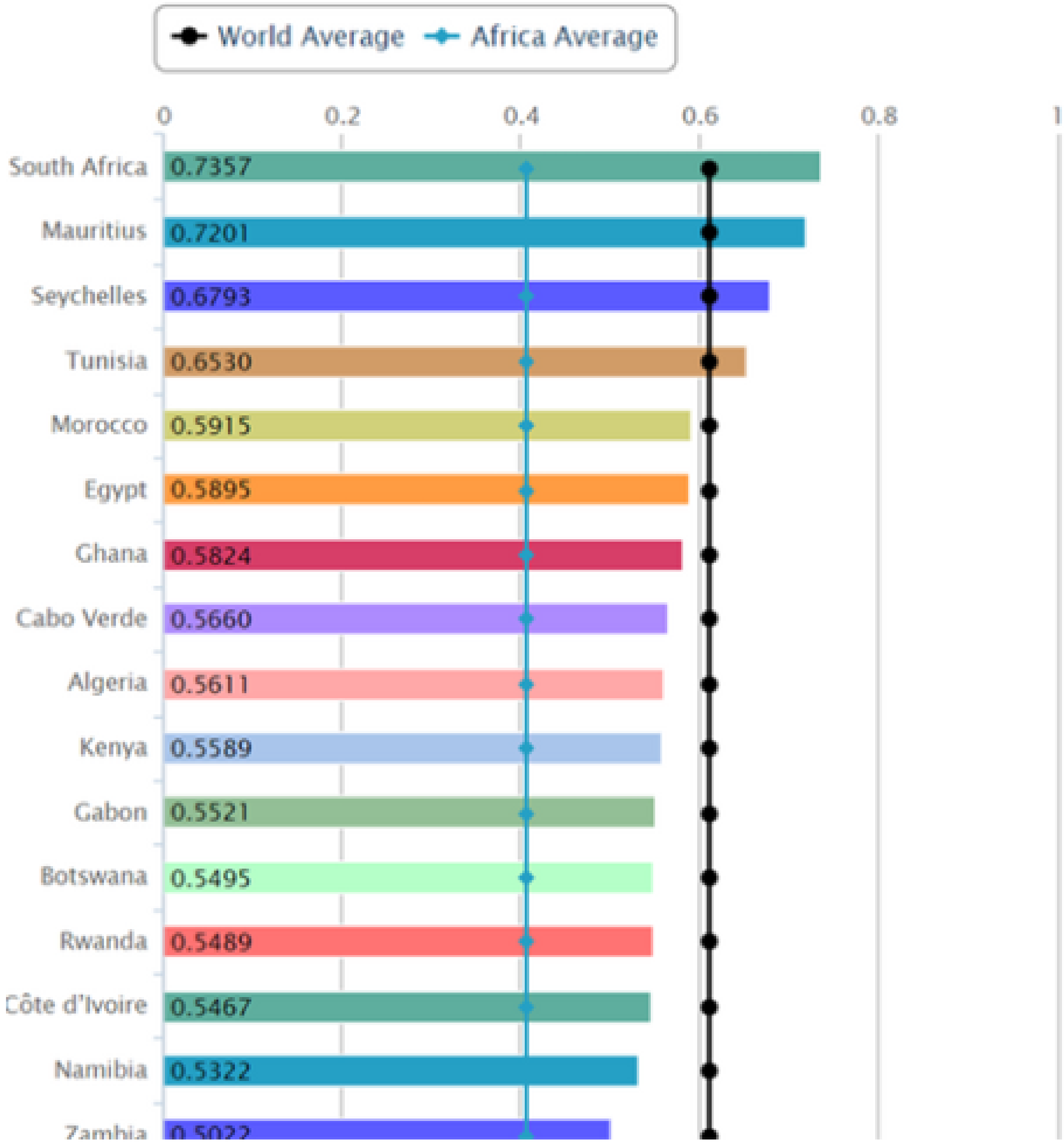

FIGURE 1.3: Algeria's Digital Development Index Ranking in Africa and the world [53]

OSI = **0.3743** Online Service Index	TTI = **0.6133** Télécommunication infrastructure index	HCI = **0.6956** Human Capital Index
EGDI = 1/3(OSI+TTI+HCI) = 1/3 (**0,3743+0,6133+0,6956**) = **0,5611**		

FIGURE 1.4: Algeria's ranking based on EGDI index [53]

1.8 Challenges in E-Government

Despite the benefits of e-government, such initiatives, particularly in developing countries, face significant challenges [20, 58]. The complexity and diversity of e-government projects suggest that there are many barriers and difficulties in their management and implementation[4]. As a result, many publications have listed a variety of challenges and issues that can hinder the adoption of e-government [4, 20, 45, 58, 59]. It can be seen that the most common challenges are represented in five categories: technical challenges, organizational challenges, social issues, human issues and financial challenges. Such category include several factors. All these challenges have a direct or indirect effect on each other [59].

1.8.1 Technical Challenge

The implementation of e-government in developing countries faces some technological challenges, such as the lack of ICT infrastructure within ministries and agencies [18]. In addition to issues of interoperability, privacy, security, availability and accessibility, which constitute critical barriers to the implementation of e-government systems

1.8.1.1 ICT Infrastructure

ICT Infrastructure is concerned about a Lack of many necessary technological resources such as high-speed internet, modern devices, and reliable software platforms to support the digitization of government services [29].

1.8.1.2 Interoperability

Interoperability is a major challenge in e-government implementation, particularly in developing countries where government departments often rely on incompatible systems and technologies [42]. These differences in data formats, software, and communication protocols complicate data exchange and system integration. Interoperability is crucial for ensuring that various government sectors, such as health, education, and law enforcement, work together effectively [60]. Without it, citizens may experience fragmented services, delays, and difficulties accessing essential services, ultimately reducing the efficiency and effectiveness of the e-government system [61].

1.8.1.3 Accessibility

Accessibility in [30], refers to "how citizens access government services and information online through multiple electronic channels." Service accessibility can thus be considered an important indicator for creating "citizen-centric" e-government services.

1.8.1.4 Availability

Availability is the number of electronic services available to citizens 24 hours a day, 7 days a week [55]. Because of its effect on service quality, it also raises citizens' satisfaction with the e-government system. The success of e-government depends on making sure that the services that are currently offered are improved and working to expand the number.

1.8.1.5 Privacy and Security

Privacy and Security concerns are major barriers to the adoption of e-government systems [63]. The digitalization of government services often requires the collection, storage, and processing of sensitive personal data, including identification information, financial records, and health data. This raises significant concerns about the protection of citizens' privacy and the potential for data breaches.

1.8.2 Organizational Challenges

Organizational challenges related to implementing e-government include the need for strong leadership and support from senior management, resistance to change, lack of collaboration between agencies, lack of trained and qualified personnel, and regulatory or policy issues [59]. These elements can hamper the effective implementation and administration of e-government systems, slow progress, and produce scattered efforts.

1.8.3 Economic Challenges

Economic challenges are a major factor in the implementation of e-government, especially in developing nations [59]. The high costs of deploying, maintaining, and setting up computer systems make electronic administration initiatives expensive and frequently unfeasible from a financial standpoint.

The high costs associated with deploying, maintaining, and setting up computer systems make electronic administration initiatives expensive and frequently unfeasible from a financial standpoint. These expenses cover the necessary software, hardware, ongoing maintenance, and training and education for both citizens and government employees. To overcome these financial barriers, public administrations must ensure long-term financial support and sustainable financing models.

1.9 Future Directions in E-Government

Emerging technologies such as artificial intelligence (AI), blockchain, the Internet of Things (IoT), big data analytics, and cloud computing are revolutionizing e-government and redefining how services are delivered to citizens. AI automates routine tasks, improves decision-making, and delivers personalized services [41], while blockchain enhances the transparency and security of government transactions. IoT enables real-time data collection and device connectivity, thereby improving public service delivery. Big data analytics provides valuable insights for informed policymaking, and cloud computing offers scalable and cost-effective solutions for government IT infrastructure [39, 43].

However, the increasing adoption of these technologies introduces several critical challenges, including heightened security risks, complex integration processes, and scalability concerns. As organizations work to harness the potential of these advanced tools, they must navigate the complexities of integrating diverse systems while ensuring their scalability to meet growing demands. A key challenge is the need for enhanced supply chain automation to boost efficiency and the necessity for robust interoperability across systems to enable seamless data exchange. In this context, Semantic Web technologies [6] offer a promising solution by providing standardized, interoperable mechanisms that facilitate smooth communication between diverse systems, thus addressing integration and scalability issues.

1.10 Conclusion

This chapter examined the evolution, importance, and key components of e-government systems, focusing on the different types of applications and their contribution to improving public sector functions. A critical analysis of e-government in Algeria reveals both the progress made in modernizing government services and persistent challenges, including infrastructure limitations, resistance to change, and barriers to digital literacy.

Furthermore, this chapter explored emerging trends, such as the integration of artificial intelligence, evolving devices, and other technological innovations, which have the potential to further improve e-government systems. However, the growing use of these technologies also highlights concerns related to security risks, integration complexity, scalability, and the demand for skilled labor. A robust technological framework and improved interoperability standards are needed to address these challenges.

The following chapters will examine how these strategies can help mitigate these problems and increase the efficiency of e-government systems, with a focus on Semantic Web services and interoperability frameworks.

Chapter 2

Web services

2.1 Introduction

Web services have revolutionized the way applications communicate over the internet, enabling interoperability between disparate systems. They allow for the integration of applications across different platforms and languages, facilitating seamless data exchange and functionality sharing. This chapter delves into the foundational concepts, architectures, applications, advantages, and challenges of web services, with a particular emphasis on their role in e-government.

2.2 Definitions

2.2.1 Web Services

Web services are defined as standardized means of interoperable communication between disparate systems over a network. They utilize protocols such as HTTP and data formats like XML and JSON to facilitate this interaction. According to the World Wide Web Consortium (W3C), web services are "a software system designed to support interoperable machine-to-machine interaction over a network" [64].

Web services are undoubtedly the most popular technology in both the industrial and academic worlds for transitioning to service-oriented architectures (SOA). The key mechanisms required for implementing an SOA are supported by a stack of standard protocols built around the definition of web services. In fact, there are two types of web services: SOAP web services and REST web services. These two types of web services offer different standards and languages to implement the principles of SOA.

2.2.2 Service-Oriented Architecture (SOA)

SOA is a widely adopted paradigm that provides a set of methods for developing and integrating systems whose functionalities are implemented as interoperable and independent services. The service-oriented architecture paradigm is generally based on a set of principles that must be followed [65, 66]:

- **Loose Coupling** refers to the level of interaction between two or more software components. Two components are considered tightly coupled if they exchange a large amount of information, whereas they are loosely coupled if they exchange minimal data. In an SOA, the coupling between client applications and services should be loose. Since web services communicate via XML-based messages, which are characterized by generality and high abstraction, they enable application cooperation while ensuring a low coupling level. As a result, a service can be modified without breaking compatibility with other services in the application.
- **Interoperability** allows applications written in different programming languages and running on various platforms to communicate with each other. By using standard formats such as XML and Internet protocols, web services ensure a high level of interoperability, regardless of the platforms they are deployed on or the programming languages used. By relying on standard message exchange formats and the ubiquity of Internet infrastructure, interoperability becomes an intrinsic characteristic of web services.
- **Reusability** helps reduce development costs by leveraging existing components. In web service-based approaches, the goal of separating operations into autonomous services is to promote reuse. When a client defines requirements, existing services can often be reused to meet part of those requirements. This approach facilitates application maintenance and significantly saves time.
- **Discovery** is a crucial step that enables service reuse. To use a service, it must first be found. The web services approach aims to minimize human intervention

to enable automatic service discovery. For instance, a developer can simply query a service search engine to find the most suitable service for their needs.

- **Composition** Multiple services can be coordinated and combined to form service compositions. This ability to build new systems from existing services is one of SOA's key advantages. Once services have been discovered, they can be assembled using Internet technologies and composition standards to create complex functionalities.

SOA is based on a model that defines interactions between three key entities: Service Provider, Service Consumer, Service Registry [67].

The interactions between these three actors follow three main communication primitives:

- Publication: Service providers register their service descriptions in the service registry (Step 1: Publication).
- Discovery: Service consumers send requests to the registry to find a service that meets their needs (Step 2: Discovery).
- Invocation: Once a suitable service is found, the client connects to the provider and uses the service based on its description (Step 3: Invocation).

This interaction model is illustrated in Figure 2.1.

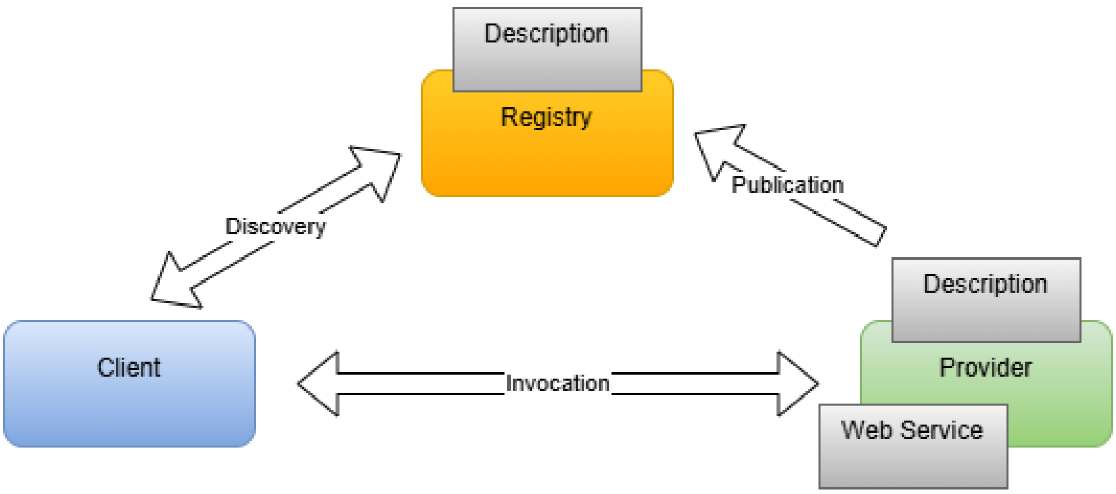

FIGURE 2.1: Interactions within a Service-Oriented Architecture

SOA aims to provide fast and easy access to functionalities offered by distributed and heterogeneous third parties. To achieve this, an environment must be established to define service interaction and usage modalities.

This environment includes a set of standards, protocols, and mechanisms, which can be categorized into two groups:

- Functional Mechanisms: These cover the core tasks involved in service usage, including publication, discovery, composition, and invocation.
- Non-Functional Mechanisms: These handle optional requirements such as security, transactions, and quality of service (QoS).

2.2.3 REST Architecture

REST (Representational State Transfer) was introduced by Roy Thomas Fielding in his dissertation, Architectural Styles and the Design of Network-based Software Architectures [68]. It is an architectural style based on a set of constraints that, when applied to system components, optimize various design criteria.

REST defines constraints for distributed hypermedia systems like the World Wide Web, aiming to enhance qualities such as task separation, simplicity, generality, and network performance. It serves as a model for how the modern web should function and, while less formal, can be compared to object-oriented or entity-relationship paradigms.

REST follows a client-server model, where the fundamental unit of information is called a resource. A resource represents any identifiable entity, such as a news article, an image, a service, or any concept. REST-based architectures rely on the following elements:

- Resource Identifier: Each resource is uniquely identified by a resource identifier. On the web, this is typically a URI (Uniform Resource Identifier).
- Resource Representation: Resources are manipulated by transferring their representations. Common formats include HTML, JSON, and XML.
- Operations on Resources: Resources are accessed and modified through a uniform interface, which uses their identifier to facilitate communication.

2.2.4 WSDL (Web Services Description Language)

WSDL is an XML-based language designed to describe the functionalities of a web service in a structured and machine-readable format. It acts as a contract between the service provider and the service consumer, detailing how a service can be accessed and used [64].

2.2.4.1 Key Features of WSDL

- Service Interface Definition: Specifies the available operations, their input/output parameters, and data types.
- Protocol and Message Format Binding: Defines how messages should be transmitted (e.g., SOAP over HTTP, REST).
- Automatic Client Code Generation: Developers can generate client stubs using WSDL, reducing manual coding efforts.
- Interoperability: Enables cross-platform communication between different systems (e.g., Java, .NET, Python).

2.2.4.2 Components of a WSDL Document

- Types – Defines the data types used in messages (commonly in XML Schema format).
- Message – Describes the input and output messages for operations.
- PortType – Defines the available operations as a collection of messages.
- Binding – Specifies the protocol used (SOAP, HTTP, etc.).
- Service – Specifies the endpoint URL where the service is accessible.

2.2.4.3 Benefits of WSDL

A standardized format for defining web services ensures consistency and interoperability across different platforms. It enables automated integration by allowing systems to seamlessly connect and interact without manual configuration. Additionally, it facilitates client code generation, reducing development time and effort. Supporting multiple transport protocols, such as SOAP, HTTP, and JMS, enhances flexibility, allowing web services to communicate efficiently across various network environments.

2.2.5 UDDI (Universal Description, Discovery, and Integration)

UDDI is a platform-independent registry that enables businesses to publish, discover, and integrate web services. It acts as a global directory that helps organizations find services based on industry classifications and service descriptions [64].

2.2.5.1 Key Features of UDDI

- Service Discovery: Enables companies to search for web services based on their name, category, or functionality.
- Business Integration: Provides a way for companies to collaborate and share services over the internet.
- Standardized Description: Uses WSDL and SOAP to describe and interact with web services.
- Decentralized Nature: UDDI registries can be public (open for all) or private (enterprise-specific).

2.2.5.2 UDDI Components

- Business Entity – Represents a business or organization providing web services.
- Business Service – Describes a specific service offered by the business.
- Binding Template – Contains the technical details (e.g., WSDL URL) for accessing the service.
- tModel (Technical Model) – Provides metadata about the service, such as supported protocols or security mechanisms.

2.2.5.3 Benefits of UDDI

The Universal Description, Discovery, and Integration (UDDI) standard offers several advantages in the world of web services. It simplifies service discovery by providing a structured registry where businesses and applications can locate services across different industries. By enabling standardized service integration, UDDI enhances interoperability, allowing systems built on diverse platforms to communicate seamlessly. Additionally, it supports automation by enabling dynamic service discovery and binding, reducing manual intervention and improving the efficiency of service-oriented architectures.

2.2.5.4 Use Cases of UDDI

UDDI plays a crucial role in various domains, enhancing the efficiency of web service management and integration. In large organizations, Enterprise Service Bus (ESB) utilizes private UDDI registries to manage internal web services, ensuring seamless communication between different applications. For B2B integration, businesses rely on UDDI to register and discover services, facilitating smooth and efficient collaboration. Additionally, in cloud computing and SaaS (Software as a Service), cloud providers use UDDI to offer and manage APIs, enabling streamlined access to diverse applications and services.

2.2.6 SOAP-based vs. RESTful Web services

There are two main ways for developing Web services: the traditional SOAP-based Web services and conceptually simpler, RESTful Web services [69]. ‘‘SOAP-based’’ Web services, also called WS/ Web services, depend on three important standardization initiatives, i.e., WSDL, SOAP, and the Universal Description, Discovery, and Integration (UDDI). Service registration, discovery and invocation are implemented by SOAP calls. SOAP-based Web services are protocol independent and stateful, but demand more computation resources, especially when handling SOAP messages. SOAP-based Web services are typically used to integrate complex enterprise applications.

By contrast, ‘‘RESTful’’ Web services utilize the REST model. REST stands for Representational State Transfer, which was introduced as an architectural style for building large-scale distributed hypermedia systems [68]. RESTful Web services are identified by

URIs, which offer a global addressing space for resource and service discovery. RESTful Web services interact through a uniform interface, which comprises a fixed set of operations in the context of the Web and the Hypertext Transfer Protocol (HTTP): GET, PUT, DELETE and POST. Services interact by exchanging request and response messages, each of which includes enough information to describe how to process the message. In contrast to SOAP-based Web services, RESTful Web services are lightweight and stateless, which are well suited for tactical, ad hoc integration over the Web. A popular technique is mashup that enables users to create situational applications based on existing application components [70].

2.3 Web Service Stack

The W3C has proposed a Web Service Stack [71], an architectural model that extends the basic or reference architecture, which defines the interactions and relationships between the three main actors (as shown in Figure 2.1).

This stack architecture accounts for more complex processes, such as service composition. It consists of multiple hierarchical layers, which is why it is referred to as a stack. Various representations and extensions of this model have been proposed in the literature [72–74].

Figure 2.2 provides a global view of these layers and the key dimensions that constitute the Web Service Stack. Each layer in this architecture relies on specific languages, protocols, and models to ensure seamless service interactions.

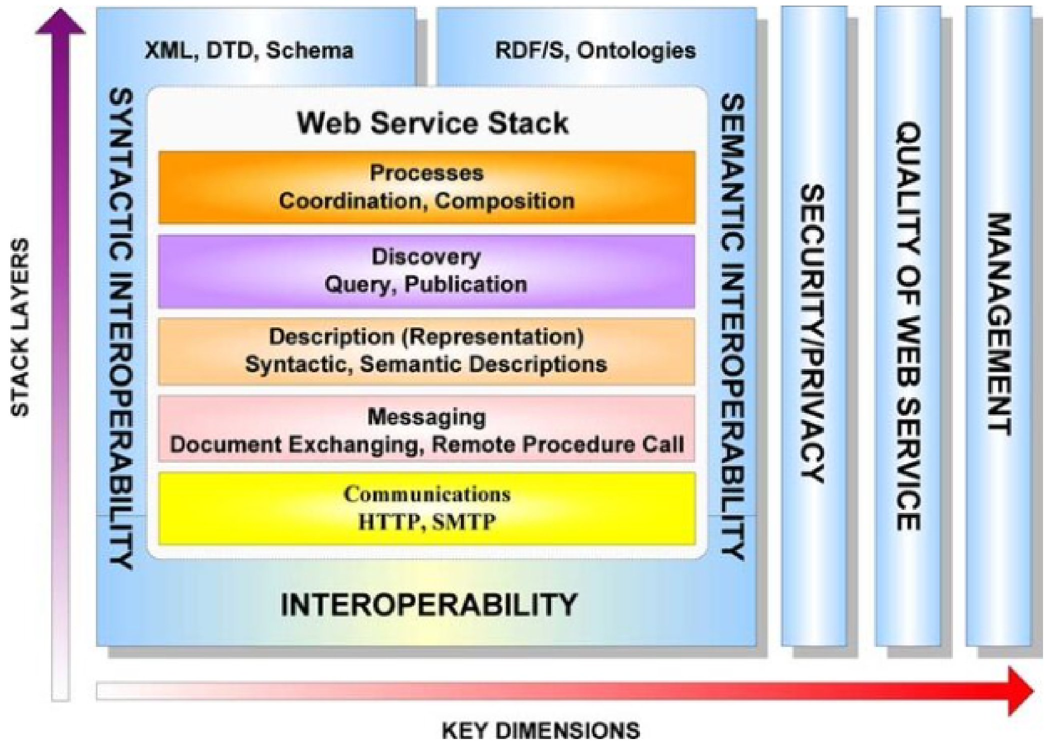

FIGURE 2.2: Web service stack and key dimensions

2.3.1 Layers of the Web Service Stack

The web service stack consists of four primary layers: communication, description, discovery, and composition. Each layer plays a crucial role in enabling seamless interaction between web services.

- **Communication Layer** This layer includes network protocols and messaging protocols that establish underlying communications. A variety of network protocols facilitate web service accessibility and message exchange. HTTP (Hypertext Transfer Protocol) is widely adopted as the standard network protocol for web services available on the internet. However, other network protocols such as SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), and JMS (Java Message Service) can also be supported. Messaging models operate with a diverse set of network protocols. For instance, SOAP (Simple Object Access Protocol) is a widely used standard that can be combined with HTTP to enable structured message exchange.

- **Description (or representation) Layer** is responsible for defining web services. It specifies their functionalities, operations, input/output data types, and binding information using a standardized service interface. The service description plays a key role in the discovery process, as it enables service requesters to locate appropriate web services based on their specifications.

- **Discovery Layer** facilitates the search and retrieval of web services. Service providers can store service descriptions in a centralized service registry, allowing service requesters to query and find services that meet their needs. This registry-based approach enhances the accessibility and organization of web services within a distributed environment.

- **Composition Layer** supports more complex interactions between web services, enabling service interoperability. It relies on the functionalities of the lower layers in the web service stack. For instance, service discovery is essential for identifying and integrating services based on their descriptions. This layer allows multiple services to be orchestrated into a service chain, forming a sequence of coordinated web services. These service compositions enhance software applications by offering richer and more advanced business functionalities.

2.3.2 Key Dimensions of Web Services

Web services encompass multiple dimensions, with interoperability, quality of service (QoS), and security being the most critical aspects.

- Interoperability: Refers to the extent to which web services can collaborate to achieve a common objective. Designed to integrate applications from distributed and heterogeneous environments [75], web services ensure seamless communication between systems. Interoperability operates on two levels: syntactic and semantic.

  - Syntactic interoperability concerns the structural aspects of web services, such as the number of parameters and their data types. XML (eXtensible Markup Language) facilitates this type of interoperability by encoding syntactic information in XML documents.

  - Semantic interoperability is more complex, as it deals with the meaning of data and services. It involves properties such as a web service's domain of interest and the functionality provided by its operations. The Semantic Web has emerged as a key solution to enhance the semantic capabilities of web services. Ontologies, for example, provide formal and explicit specifications of shared conceptualizations, helping to structure and define relevant concepts within a given domain [76].

- Quality of Service (QoS): With the growing number of web services, competition has intensified among providers offering similar functionalities. QoS is a crucial factor in differentiating competing services. [77]. It encompasses various quality parameters that define the behavior of a web service in delivering its functionalities. These parameters may include aspects such as performance, reliability, and availability, ensuring that services meet the specific requirements of different application domains.

- Security is a major concern in web service deployment. While web services facilitate interoperability, they also expose internal applications and databases to external users[78], increasing vulnerability to security threats. To mitigate risks, web service security must address key aspects such as authentication (verifying user identity), authorization (managing access permissions), confidentiality (protecting sensitive data), and integrity (ensuring that data remains unaltered). Robust security measures are essential to safeguarding web services against unauthorized access and cyber threats [76].

2.4 Applications of Web Services

Web services are widely used across multiple industries, enabling seamless integration, automation, and data exchange. Their interoperability, scalability, and real-time communication capabilities make them essential in various domains.

- **E-government** Refers to the use of information and communication technologies (ICTs) to optimize government operations and deliver more efficient services to citizens, businesses, and government agencies. By leveraging digital solutions, e-government enhances transparency, accessibility, and responsiveness in public administration [79].

  Web services play a crucial role in enabling interoperability between various government systems. They facilitate secure data exchange across agencies, ensuring seamless communication and collaboration. Additionally, web services improve the efficiency of public services by providing a centralized platform where citizens can easily access government information and perform essential tasks such as online tax filing, license renewal, and e-voting.

A notable example of e-government implementation is Taiwan's digital governance initiatives, which demonstrate how web services enhance public service delivery. By integrating disparate systems and enabling real-time data exchange, Taiwan has streamlined government operations, improving service efficiency and citizen engagement.

- **E-commerce platforms** depend on web services to efficiently handle transactions, manage inventories, and enhance the overall customer experience. By leveraging web technologies, businesses can streamline operations, ensure secure transactions, and provide seamless shopping experiences across multiple platforms [80].

  Web services play a critical role in facilitating secure transactions, such as payment processing through APIs like PayPal and Stripe. They also automate inventory management, ensuring real-time stock updates across different sales channels. Additionally, web services enhance customer service by integrating Customer Relationship Management (CRM) systems, helping businesses track customer interactions and personalize services. Another key function is cross-platform integration, which allows major online marketplaces like Amazon, Shopify, and eBay to efficiently communicate and share data.

  A leading example of web services in e-commerce is Amazon, which utilizes them to integrate payment gateways, recommend products based on user behavior, and connect third-party seller APIs. These integrations enhance operational efficiency and provide customers with a seamless shopping experience.

- **The healthcare industry** greatly benefits from web services, as they facilitate interoperability between electronic health records (EHRs), medical devices, and healthcare applications. By enabling seamless data exchange, web services improve patient care, enhance operational efficiency, and support emerging healthcare technologies.

Web services ensure the secure and efficient exchange of patient data between hospitals, clinics, and insurance providers, reducing administrative burdens and improving patient outcomes. They also enhance real-time access to medical records, minimizing errors and delays in diagnosis and treatment. Another key advantage is their role in telemedicine, allowing doctors to consult with patients remotely, providing access to medical expertise regardless of location. Additionally, web services help integrate wearable health devices with patient health records, enabling real-time monitoring of vital signs and early detection of health issues.

A significant example of web services in healthcare is the Fast Healthcare Interoperability Resources (FHIR) standard, which facilitates secure and standardized patient data exchange. By leveraging web services, FHIR ensures that different healthcare systems can communicate effectively while maintaining data security and compliance with regulatory standards.

- **The Internet of Things (IoT)** connects smart devices and enables data exchange, automation, and real-time monitoring across various industries. By leveraging IoT, businesses and governments can improve efficiency, optimize resource utilization, and enhance user experiences [81].

  Web services play a critical role in enabling communication between IoT devices using lightweight protocols such as RESTful APIs and MQTT. These protocols ensure seamless data exchange between connected devices in smart homes, industrial automation, and connected vehicles. Additionally, web services facilitate real-time data collection and processing, which is essential for monitoring systems and making data-driven decisions.

  Another key function of web services in IoT is enhancing automation by integrating IoT devices with cloud platforms, allowing centralized control and analytics. Security is also a major concern in IoT ecosystems, and web services help ensure secure remote access through authentication mechanisms like OAuth and JWT, preventing unauthorized access to sensitive data.

  A prime example of web services in IoT is their application in smart cities. IoT-enabled web services support traffic management systems, environmental monitoring, and smart energy grids, allowing cities to optimize traffic flow, reduce pollution, and improve energy efficiency. These advancements lead to more sustainable and efficient urban environments.

- **Cloud Computing** Web services are the foundation of cloud computing, providing on-demand access to computing resources over the internet. They enable businesses and individuals to leverage cloud infrastructure without the need for physical hardware, reducing costs and increasing efficiency [81].

  Web services facilitate communication between cloud-based applications, ensuring seamless data exchange and integration. They support various cloud service models, including Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). These models allow businesses to adopt cloud-based solutions for application hosting, development, and infrastructure management.

  One of the key advantages of web services in cloud computing is elastic scalability. Organizations can dynamically scale their IT infrastructure based on demand, optimizing performance while minimizing costs. This flexibility makes cloud computing a preferred choice for businesses of all sizes.

  Leading cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud rely on web services to manage and provision cloud resources. These platforms use web services to deliver virtual servers, storage, databases, and networking solutions, empowering businesses to build and deploy applications efficiently.

- **Financial Services and Banking** The financial sector relies on web services to facilitate secure and efficient transaction processing, account management, and regulatory compliance. By enabling seamless communication between financial institutions, web services enhance customer experience, automate financial operations, and improve security.

  Web services play a crucial role in secure online banking and payment processing, ensuring that users can transfer funds, pay bills, and manage accounts safely. They also support automated stock trading and risk analysis, allowing financial firms to make data-driven investment decisions in real time.

  Another significant advantage is fraud detection and security integration. Web services connect banking systems with AI-powered security solutions, helping to identify and prevent fraudulent activities. Additionally, open banking APIs enable third-party applications to interact with banking systems, fostering innovation in financial technology (FinTech).

Prominent financial platforms such as PayPal, VisaNet, and SWIFT leverage web services to process transactions in real time, ensuring secure and efficient financial operations across global markets.

- **Enterprise Resource Planning (ERP)** Enterprise Resource Planning (ERP) systems integrate and streamline various business processes, including finance, human resources, supply chain management, and customer service. By centralizing data and automating workflows, ERP solutions enhance operational efficiency and decision-making within organizations.

  Web services play a vital role in real-time synchronization between different business applications, ensuring that data remains consistent across departments. They also enhance decision-making by providing real-time analytics, allowing businesses to gain insights into operations, track performance, and optimize processes.

  Furthermore, web services facilitate third-party integrations with essential business tools such as Customer Relationship Management (CRM) systems, HR management platforms, and inventory management software. This interoperability enables organizations to expand their ERP capabilities and tailor solutions to their specific needs.

  Leading ERP providers like SAP and Oracle leverage web services to connect internal business applications, enabling seamless communication between various enterprise functions and improving overall business agility.

- **Social Media and Communication** Social media platforms rely on web services to enhance user engagement, content sharing, and real-time communication. By integrating various functionalities, web services enable seamless interactions between users and third-party applications, making social media more dynamic and interactive.

  Web services facilitate API integrations, allowing platforms to exchange data efficiently. For example, embedding Twitter feeds on websites or enabling cross-platform sharing between different social networks. They also power messaging and video conferencing services, such as WhatsApp, Zoom, and Microsoft Teams, ensuring smooth and secure communication.

Additionally, web services support real-time notifications through WebSockets and push APIs, keeping users updated with instant alerts, messages, and content updates. These features enhance user experience by providing timely and interactive engagements.

Major social media platforms like Facebook and Twitter leverage web services to enable third-party app integrations, allowing developers to create applications that interact with their ecosystems, expanding functionalities and user engagement across different digital platforms.

2.5 Architecture of Web Services

The architecture of web services is built on three key components that work together to facilitate service discovery, invocation, and communication [82]. The service provider is responsible for creating, maintaining, and hosting the web service, ensuring its accessibility to clients over the internet. It also defines the service description and publishes it in a service registry for discovery. The service requester, or client, is the application or system that consumes the web service. It locates the necessary service through the registry and establishes a connection to invoke it. The service registry serves as a directory where service providers publish their web services, allowing requesters to search, discover, and integrate available services. A widely used standard for managing service registries is UDDI (Universal Description, Discovery, and Integration), which streamlines service discovery and integration. Together, these components form the foundation of web service architecture, enabling seamless communication and interoperability across different systems.

2.6 Benefits of Web Services

Web services offer numerous benefits that make them an ideal choice for system integration [83]. One of the key advantages is interoperability, as web services facilitate communication between applications built on different platforms and programming languages. This ensures seamless integration in heterogeneous environments. Another significant benefit is reusability, allowing web services to be leveraged across multiple applications and systems. This reduces both development time and costs, making software solutions more efficient. Additionally, web services provide scalability, enabling systems to handle varying workloads and support an increasing number of users and transactions without compromising performance. Furthermore, security is a critical aspect of web services, with protocols like WS-Security ensuring secure communication through message integrity, confidentiality, and authentication mechanisms. These advantages collectively make web services a powerful tool for modern software development and system interoperability.

2.7 Challenges of web services

Despite their advantages, web services face several challenges [64]:

- Performance Overhead: Web services, particularly SOAP-based ones, can suffer from performance overhead due to the verbosity of XML and the need for parsing.
- Scalability Issues: Handling high levels of concurrent requests can be challenging without adequate infrastructure, especially for stateful services.
- Security Concerns: Ensuring secure communication, preventing unauthorized access, and safeguarding data from cyber attacks are critical challenges.
- Interoperability Constraints: Although web services aim to be platform-independent, differences in implementation and standards can lead to compatibility issues.
- Lack of Standardization: Inconsistent implementation of protocols such as SOAP and REST across platforms can hinder seamless integration.

2.8 Conclusion

Web services are a cornerstone of modern software development, enabling interoperability and integration across diverse platforms and applications. They have proven invaluable in domains such as e-government, healthcare, and e-commerce, streamlining processes and improving efficiency. However, challenges such as performance, security, and standardization need to be addressed to fully harness their potential. With ongoing advancements, web services are poised to play an even more significant role in the future of technology integration. In the next chapter, we will explore key concepts related to semantics in web services, the role of ontologies, and web service decomposition, providing a deeper understanding of how these elements contribute to more intelligent and adaptive service architectures.

Chapter 3

Semantic and Composition in Web Services

3.1 Introduction

Integrating semantic technologies with web services has brought significant advancements to distributed computing. Semantic web services enable systems to interact more intelligently and autonomously by incorporating meaning into data, thus enhancing interoperability, automation, and scalability. Additionally, decomposing web services into smaller, semantically enriched components offers numerous benefits, such as improved modularity and reuse. This chapter delves into the concepts of semantics in web services and the composition of these services, with a particular focus on their applications, tools, and future directions.

Part 1: Semantic in Web Services

3.2 Definition of Semantic in Web Services

The Semantic Web initiative, spearheaded by the World Wide Web Consortium (W3C) with contributions from numerous researchers and industry partners, has gained significant traction in recent years. This initiative has led to the development of technologies, frameworks, and tools aimed at addressing key limitations of current web technologies. Notably, the research community shares a broad consensus on the vision of the Semantic Web and the core technologies that define it.

According to the free encyclopedia Wikipedia, the Semantic Web is described as "a project aimed at creating a universal medium for information exchange by assigning computer-understandable meaning to web content" [84].

The World Wide Web Consortium (W3C) highlights the importance of XML and the Resource Description Framework (RDF) as foundational technologies, stating that "the Semantic Web provides a common framework for sharing and reusing data across applications, enterprises, and communities. It is built on the Resource Description Framework, which integrates various applications using XML for syntax and URIs for naming" [85].

Tim Berners-Lee emphasizes that the Semantic Web should not be viewed as a separate development from the existing web. In his well-known Scientific American article, he explains that "the Semantic Web will introduce structure to the meaningful content of web pages, creating an environment where software agents can navigate from page to page and efficiently perform complex tasks for users. It is not a distinct Web but an extension of the current one, where information is assigned well-defined meaning, enabling better collaboration between computers and humans" [86].

Dumbill emphasizes that the core vision of the Semantic Web is to manage complexity by automating tasks that are currently performed manually. He states that "the fundamental goal of the Semantic Web is to make web information easily processable by computers, ultimately enhancing the web's effectiveness for users. This improvement is achieved by automating or facilitating tasks that are presently challenging, such as locating content, aggregating and cross-referencing information, and drawing conclusions from data found in multiple sources" [87].

The Semantic Web is an extension of the existing web, where information is precisely defined using well-structured vocabularies that can be understood by both humans and computers. This technology relies on the Resource Description Framework (RDF), a W3C standard for describing and structuring data.

A key concept in the Semantic Web is Ontology, which provides a formal representation of knowledge within a specific domain. The Web Ontology Language (OWL) and RDF are the two W3C-recommended models for ontology representation. Ontology is defined as an explicit specification of conceptualization, encompassing a structured model of a domain with possible constraints. This conceptualization is expressed using a specific modeling language and terminology. While ontology defines a domain, a knowledge base built on ontology represents a particular state of affairs within that domain.

Various methodologies for ontology development have been proposed in the literature, differing in the steps and tasks involved. In the e-government domain, defining the scope and coverage of an ontology is crucial, as government services often interconnect within the same department or across multiple departments.

Ontology engineering follows a three-layered development approach, transitioning from human-readable forms to machine-processable structures. These layers include:

- Informal Ontology: A conceptual representation understandable by humans.
- Semi-Formal Ontology: A structured representation that introduces logical relationships.
- Formal Ontology: A fully defined and machine-processable ontology.

The domain ontology, serving as the foundational model for formal ontology development, requires semantic consistency evaluation. This is achieved by creating a semi-formal representation of the domain ontology using Description Logic (DL). Common formalisms for representing semi-formal ontologies include UML class diagrams and Description Logic.

The Unified Modeling Language (UML) is a standardized, general-purpose modeling language widely used in object-oriented software engineering. It provides a set of graphical notation techniques to create visual models of complex, software-intensive systems. UML integrates various modeling approaches, including:

- Data modeling (e.g., Entity-Relationship Diagrams).
- Business modeling (e.g., workflows).
- Object modeling (e.g., class diagrams).

- Component modeling (e.g., system architecture).

It is applicable across all stages of the software development lifecycle and is compatible with multiple implementation technologies, making it a versatile tool for software design and system analysis.

The Unified Modeling Language (UML) provides a standardized approach to visualizing a system's architectural blueprints, encompassing various elements such as:

- Activities
- Actors
- Business processes
- Database schemas
- Logical components
- Programming language statements
- Reusable software components

Description Logic (DL) is a formal language used for knowledge representation, employing mathematical logic symbols such as subset, union, intersection, universal and existential quantifications to define relationships between different elements within a domain.

The Semantic Web offers several languages for formal ontology representation, including XML, RDF, and OWL. OWL (Web Ontology Language) plays a crucial role in service composition, searching, matching, mapping, and merging within e-government systems. It enhances integration, maintenance, and interoperability among government services.

Generating OWL ontology from a government service domain is a fundamental step in advancing Semantic Web applications, particularly in e-government. This process facilitates automated service discovery, efficient data exchange, and streamlined digital governance.

3.3 Advantages of Semantic Web in E-governance

The Semantic Web provides a robust foundation for achieving a knowledge-based, user-centric, distributed, and interoperable e-government system. Unlike traditional Web Services, where users must think in terms of data, Semantic Web technologies shift the focus to services, allowing for a more intuitive and user-friendly interaction with government platforms. However, despite their advantages, traditional Web Services face significant limitations that hinder their effectiveness in dynamic and complex environments such as e-government.

One of the primary challenges of traditional Web Services is their inability to enable automatic service discovery, composition, and selection. These processes require human intervention, making the integration of services labor-intensive and less efficient. As e-government services expand in scope and complexity, the need for automated service management becomes increasingly critical. This limitation is addressed by Semantic Web technologies, which enhance machine-processable service descriptions, enabling greater automation. The fusion of Semantic Web technologies with Web Services has led to the emergence of Semantic Web Services (SWS), a paradigm designed to facilitate seamless, intelligent, and automated service interaction.

A key benefit of Semantic Web Services is their ability to aggregate and reuse diverse information resources cost-effectively. This aggregation supports transparent interactions between government entities, community organizations, and individual citizens. By enabling efficient data integration, SWS enhances decision-making and service accessibility, making government services more responsive to public needs. Several researchers, including [88, 89], have explored ontology-based service integration within e-government. Their studies propose domain-specific ontologies that enforce semantic integration and interoperability, allowing different government agencies to seamlessly share and process information.

The role of ontology in e-government is further emphasized by Lijiljana Stojanovic et al. [90], who highlight how semantic technologies improve the management of changes in government services. Unlike conventional data processing approaches, which focus on storing and retrieving raw data, semantic technologies add meaning to the data, allowing computers to interpret, process, and integrate information intelligently. This capability ensures that government systems remain adaptable, efficiently handling changes in policies, regulations, and service requirements while maintaining high levels of interoperability.

In conclusion, Semantic Web Services represent a transformative approach to e-government service integration. By leveraging ontology-driven models, intelligent automation, and semantic interoperability, governments can enhance service delivery, improve efficiency, and offer more adaptive and user-friendly digital platforms. As research in this field continues to evolve, the adoption of Semantic Web technologies in e-government will play a crucial role in shaping the future of smart and interconnected public services.

3.4 Ontology

The concept of ontology has been defined in various ways throughout the literature. One of the most relevant definitions, particularly in the context of our work, is that of Gruber, who describes ontology as ”an explicit specification of a conceptualization.”

Ontology is also defined as ”a formal and implicit specification of a shared conceptualization” [91]. Essentially, it serves as a knowledge representation framework that structures vocabulary and enables its use based on a well-defined consensus.

An ontology that includes instances of individual classes constitutes a knowledge base. The class is the core element of any ontology, and it can have subclasses that provide a more detailed representation of the superclass concept. The relationship between a subclass and its superclass is known as a subsumption or ”is-a” relationship.

Beyond subsumption, various other types of relationships can be defined depending on the ontology's design context. Additionally, ontology classes are characterized by properties, which are classified into two main types:

- Data properties, which define attributes of a class.
- Object properties, which establish relationships between different ontology classes.

In computer science, ontologies are developed in various fields such as knowledge engineering, artificial intelligence, information systems management, and the Semantic

Web. In all these domains, conceptualization plays a fundamental role in structuring and organizing information.

3.4.1 Types of Ontologies in The Semantic Web

The Semantic Web literature identifies several types of ontologies, with the most common being those described by [92]:

- Generic Ontology: Describes high-level concepts such as space, time, matter, objects, events, and actions, which are independent of any specific problem or application domain.
- Domain Ontology: Defines a specialized vocabulary for a specific domain, refining concepts from generic ontology (e.g., electronics, automotive industry).
- Task Ontology: Focuses on the vocabulary associated with a specific task or activity.
- Method Ontology: Clarifies the role of each concept by providing detailed arguments on its usage.
- Application or Task-Domain Ontology: Represents a combination of domain and task ontologies, defining concepts specific to both a domain and a particular task, which are often derived from related ontologies.

We can potentially illustrate the hierarchical relationships between the different types of ontologies in the figure 3.1below.

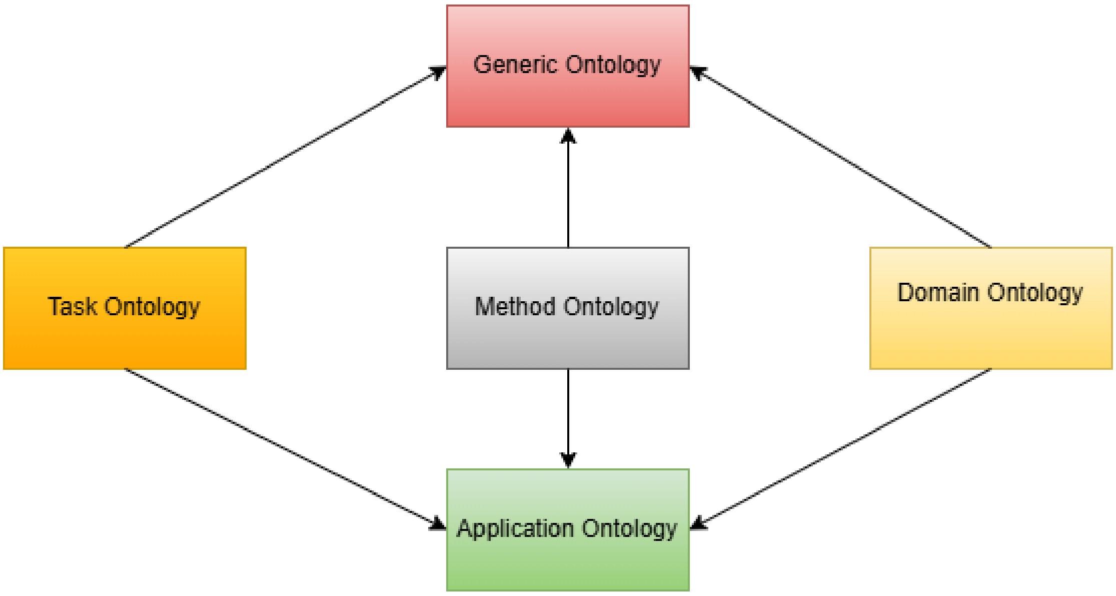

FIGURE 3.1: Relationship between Types of Ontologies

3.4.2 Representation Languages and Construction Tools

3.4.2.1 Ontology Representation Languages

The implementation of an ontology is based on the construction of a formal model represented in an ontology language. There are primarily two categories of ontology representation languages:

- Classical languages
- Markup languages

For the first category of ontology representation languages, the following examples can be cited:

- Ontolingua: Designed for portable ontologies, it defines classes, relations, and functions based on KIF (Knowledge Interchange Format) formalism [93]. Ontolingua can translate generic ontologies into domain-specific applications. It is a declarative language built on first-order predicate logic.

- Loom: A high-level programming language and environment used for building expert systems and other intelligent applications. It leverages rule-based inference, semantic unification, and object-oriented technologies to provide deductive support.
- OIL (Ontology Inference Layer): A formal ontology language that combines formal semantics with descriptive reasoning methods. It is widely used on the Web and is based on RDF/RDFS and XML formalism [94].
- SHOE (Simple HTML Ontology Extensions): A markup language that extends HTML to allow web authors to annotate their documents in a way that is machine-readable. This enables automated agents to interpret and process web content effectively [95].

For the second category, there are numerous languages and formalisms used for ontology representation. The W3C has proposed the ontology representation language pyramid, as shown in the figure 3.2 below.

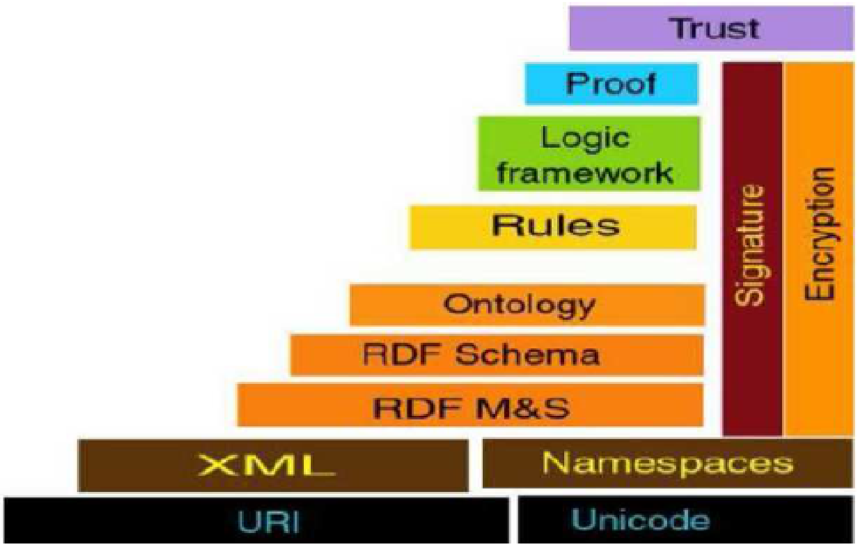

FIGURE 3.2: Semantic Web Architecture Stack

However, only a few languages have successfully represented ontologies on the web, including:

- XML (eXtensible Markup Language): A markup language that provides a structured syntax for describing, creating, and manipulating documents. It utilizes namespaces to distinguish elements and ensure consistency. XML documents are validated using schemas; however, XML does not impose semantic constraints, meaning it lacks the ability to provide machine-understandable meaning to the content. As a result, XML alone is insufficient for enabling software to comprehend the semantics of included data.

- RDF (Resource Description Framework): A semantic description language developed by W3C as a complement to XML. RDF provides a standardized and interoperable mechanism for describing resources without making assumptions about document structure or context. RDF primarily consists of three components:

  - Resources – Identified by a web address (URI).

  - Properties – Attributes describing these resources.

  - Objects – Values assigned to the properties of a given resource.

- OWL (Web Ontology Language): A knowledge representation language specifically designed for web ontologies, built on RDF's data model. OWL is a W3C-recommended standard and is available in three increasingly expressive sublanguages:

  - OWL Lite – A simplified version for basic ontologies.

  - OWL DL (Description Logic) – Ensures decidability and supports reasoning.

  - OWL Full – Offers maximum expressiveness but lacks computational guarantees [174].

  OWL also includes a dedicated query language, allowing more advanced interactions with ontologies.

- SPARQL (SPARQL Protocol and RDF Query Language): A query language for RDF-based data, combining XML's structural capabilities with formal semantics that machines can process efficiently [184].

For the implementation of the IT Governance Ontology, OWL was chosen due to its richness, expressiveness, and interoperability with other ontology representation languages, making it the most suitable choice for achieving semantic integration and reasoning.

3.4.2.2 Ontology Construction Tools

There are two main categories of ontology construction tools:

- Traditional Tools: These tools focus on the symbolic level of ontology construction (e.g., Ontolingua Server). They enforce specific languages and structures, requiring ontology creators to adhere to predefined syntax and frameworks.
- Knowledge-Centric Tools: These tools emphasize the knowledge level, allowing ontology creation independently of the representation language. The operationalization process is handled automatically, enabling greater flexibility in ontology development and implementation.

Several tools are widely used for ontology construction, including:

- Protégé [96]: A graphical environment for ontology development based on the hierarchical knowledge model (classes, attributes, and properties). It is one of the most popular ontology editors, supported by a large user community. Protégé offers compatibility with OWL, knowledge base management, ontology visualization, as well as merging and integration capabilities.
- OILEd [97]: A platform focused on class hierarchy, providing role specialization and definition testing. However, it is limited to the construction of OIL ontologies.
- OntoEdit [98]: A proprietary ontology development tool that supports hierarchical concepts and axiom representation. However, its functionality is restricted to lexical comparison of terms, limiting its flexibility for broader ontology applications.

3.4.2.3 Ontology Development Approaches and Methodologies

In the Semantic Web literature, there is no unified life-cycle model for ontology development methodologies and techniques. However, researchers have identified three primary approaches for designing a formal ontology [99]:

- Bottom-Up Approach: Starts from specific instances and generalizes concepts progressively.

- Top-Down Approach: Begins with high-level abstract concepts and refines them into more specific entities.
- Middle-Out Approach [100]: Balances both top-down and bottom-up strategies to construct ontologies more efficiently.

Later, Gomez-Perez [101] introduced METHONTOLOGY, a widely recognized ontology development methodology. It follows a structured lifecycle, including:

- Specification: Defining the ontology's purpose and scope.
- Knowledge Acquisition: Collecting relevant domain knowledge.
- Conceptualization: Structuring the information into ontology models.
- Integration: Merging with existing ontologies when applicable.
- Implementation: Translating the conceptual model into an ontology representation language.
- Evaluation: Assessing the ontology's consistency and correctness.
- Maintenance: Updating and refining the ontology over time.
- Documentation: Recording the ontology development process for future use.

Other ontology development methodologies also exist, such as TOVE [102], which is specifically designed for ontology evaluation based on predefined competency scenarios. Additionally, some researchers have proposed ontology acceptance criteria [103], focusing on the challenges and difficulties in ontology construction rather than defining a full lifecycle methodology.

Part 2: Composition of Web Services

3.5 Foundational Concepts of Web Service Composition

Web Service Composition (WSC) plays a vital role in integrating distributed and heterogeneous systems, facilitating the development of complex and value-added applications.

By leveraging existing services, organizations can streamline development, minimize costs, and quickly adapt to evolving business demands. With the rapid growth in the number of available web services, the need for efficient and scalable composition techniques has become increasingly critical. This section explores the latest methodologies, tools, and challenges in WSC while highlighting emerging trends and future research directions in the field.

3.5.1 Service Composition

Service composition serves as a mechanism for assembling and coordinating existing services to create value-added applications. This approach offers significant benefits, including cost reduction and lower development effort. Moreover, the composite services generated through this process can themselves function as building blocks for future compositions, further enhancing reusability and efficiency. The service composition process consists of several key phases, as illustrated in Figure 3.3 [104]:

- Composition Planning: The first phase involves defining the desired service and breaking it down into a series of abstract tasks.
- Service Discovery: This phase focuses on identifying candidate services that meet the functional and non-functional requirements of each task in the composition.
- Service Selection: Once potential services are discovered, the best-suited services are chosen based on user requirements, ensuring optimal performance for each task.
- Service Execution: In the final phase, the selected services are invoked and executed sequentially, ultimately delivering the fully composed service.

This structured approach ensures that service composition remains efficient, adaptable, and scalable, allowing organizations to optimize their service-oriented architectures.

For over a decade, WSC has been a dynamic field of research and development. Various composition approaches have been proposed, each differing based on specific parameters [104–108] (Figure 3.4). These classifications help in understanding the strengths and limitations of different methods, paving the way for more efficient and scalable composition techniques.

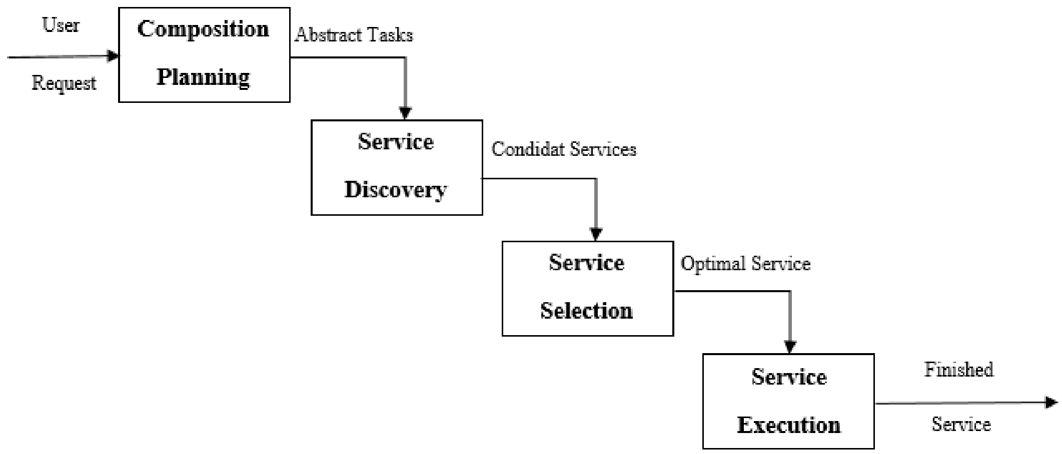

FIGURE 3.3: Process of Service Composition

WSC approaches can be categorized based on different parameters, including user involvement, composition strategy, execution control, and composition type:

- Based on Composition Manner: This classification considers the level of user involvement in defining the composition scheme.

  Approaches can be:

  - Manual: The user explicitly defines the composition.
  - Semi-automatic: The system assists the user in creating the composition.
  - Automatic: The composition is generated without user intervention.

- Based on Composition Strategy: This classification depends on when service discovery and workflow specification (executable plan) occur.

  Approaches can be:

  - Static: The workflow is predefined before execution (a priori).
  - Dynamic: The composition is determined at runtime, after receiving the user's request.

- Based on Composition Control: This classification focuses on how the composed web services (WS) are executed.

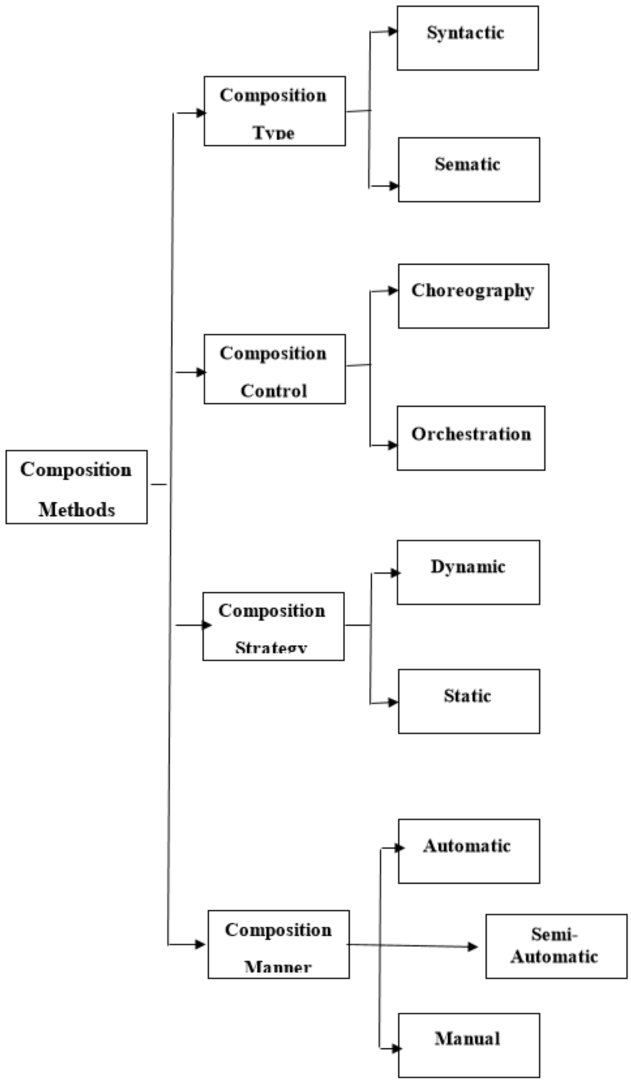

FIGURE 3.4: Classification of composition methods

Approaches include:

- Orchestration: A central entity manages and controls the execution of services.
- Choreography: Each service executes autonomously, following predefined interaction rules.

- Based on Composition Type This classification is based on the functional capabilities of services, which may include inputs, outputs, service category, behavior, annotations, and user preferences.

Approaches can be:

- Syntactic (Non-Semantic) Approaches: Focus on structural compatibility without considering meaning.
- Semantic Approaches: Use ontology and formal descriptions to enable intelligent service matching and integration.

These classifications provide a structured understanding of WSC methodologies, helping to identify the most suitable approach for different scenarios and application domains.

3.5.2 Frameworks and Tools for WSC

The implementation of Web Services (WS) and their compositions requires a combination of programming languages, frameworks, and tools to facilitate service development, integration, and orchestration. In service-based systems, the composition process (i.e., the ordering of service calls in orchestration and message exchanges in choreography) must be defined using a WSC language.

To achieve service selection, invocation, execution control, and service compensation, system designers must utilize WSC platforms. Several studies in the literature provide comparative analyses of WSC standards and platforms [104, 105, 109, 110]. Given the large number of existing standards, we focus on key tools commonly used for WS implementation and composition.

To assist developers in selecting the most suitable framework based on their needs, requirements, and preferences, we consider the following key features:

- Automation Level: Describes the composition mode based on the level of user involvement in defining how services are coordinated to fulfill a request.

  Composition can be:

  - Automatic: Specifications are generated by an application implementing a composition algorithm.
  - Semi-automatic: The user interacts with a tool to guide the composition process.
  - Manual: The user fully specifies the composition without automated assistance.

- Composite Service Modeling: Refers to the methods and formalisms used to describe the composite service process.
- Non-Functional Properties (NFPs): Indicates whether the framework supports the definition of NFPs (”Yes” or ”No”).
  - Quality of Service (QoS) is typically used to define non-functional requirements such as performance, availability, and security.
- Composition Correctness: Refers to the capability of the platform to verify the correctness and reliability of the composite service against defined requirements (”Yes” or ”No”).
- Service Selection: Indicates whether the framework provides syntactic or semantic support for automated service matching and selection.
- Execution Monitoring: Describes whether the composite web service execution engine operates in a centralized or decentralized manner.

By evaluating these features, developers can select the most appropriate framework for implementing web services and their compositions, ensuring efficiency, scalability, and adaptability in service-oriented architectures.

3.5.3 Static or Dynamic Composition

Static composition involves creating a fixed and predetermined arrangement of WS during design-time or deployment-time. In this approach, the composition logic is explicitly defined and hardcoded into the application code, remaining unchanged throughout the application’s execution.

Characteristics and Advantages:
- Best suited for applications with stable and well-defined requirements, where service interactions and data flows are known in advance.
- Since the composition plan is predetermined, performance and efficiency can be optimized, as there is no need for runtime decision-making.

Limitations:
- Lacks flexibility and adaptability, making it difficult to accommodate dynamic changes

Table 3.1: Key Features for Describing Web Service Frameworks

Service	**Automation Level**	**Composite Service Modeling**	**Non-Functional Properties**	**Service Selection**	**Adaptability**	**Execution Engine**	**Correctness of Checking**
eFlow	Semi-Automatic	Self-defined Graph	No	Syntactic	Yes	Centralized	No
Self-Serv	Semi-Automatic	Formal Method	No	Syntactic	Yes	Decentral-ized	No
METEOR-S	Semi-Automatic	Standard Language	Yes	Semantic	Yes	Centralized	No
FUSION	Automatic	Planning Language	No	Syntactic	No	Centralized	No
WebDG	Semi-Automatic	Self-defined Graph	Yes	Semantic	No	NA	Yes
AO4BPEL	Semi-Automatic	Standard Language	Yes	Semantic	Yes	Centralized	Yes
OWLS-Xplan	Automatic	Planning Language	Yes	Semantic	No	NA	No

in service availability, user requirements, or system environments.
- Not suitable for applications requiring real-time adaptability, such as IoT or cloud-based systems.

Dynamic composition, in contrast, takes place at runtime, enabling the on-the-fly creation of complex service compositions while incorporating real-time service updates.

Characteristics and Advantages:
- Ideal for applications with evolving requirements, such as IoT environments, where service availability and capabilities may change over time.
- Ensures greater flexibility, allowing services to be selected and composed dynamically based on current conditions.

Limitations:
- Not always feasible for scenarios where the composition scheme is only partially defined or cannot be pre-determined.
- Potential overhead in processing and decision-making due to the need for real-time service discovery and selection.

By understanding the trade-offs between static and dynamic composition, developers can choose the most appropriate approach based on the specific application requirements and operational constraints.

3.5.4 Orchestration or Choreography Composition

Orchestration is a centralized approach to web service composition, where a central coordinator, known as the orchestrator, manages and controls the interactions between web services. The orchestrator is responsible for:

- Coordinating the execution flow of services within the composition.
- Managing data exchange between services.
- Sequencing service calls to ensure correct execution order.
- Handling error management and compensation when failures occur.

In this approach, individual services are unaware that they are part of a larger business process and do not interact autonomously; instead, they rely on the orchestrator to direct their interactions.

Choreography, on the other hand, is a decentralized approach where services interact directly with each other without a central controller. Each participating service is responsible for:

- Understanding its role in the composition.
- Determining when and how to exchange messages with other services.
- Cooperating dynamically based on predefined rules and changing conditions.

Unlike orchestration, choreography is better suited for adaptive and dynamic environments, where services must collaborate flexibly in response to real-time events and varying conditions.

3.5.5 Syntactic or Semantic Composition

Syntactic composition constructs the composition workflow by considering only the syntax of WS descriptions. The dependencies between services are managed based on their input and output parameters, which are specified in the WSDL (Web Services Description Language) file [111]. This approach ensures structural compatibility between services but does not account for the meaning of the exchanged data.

Semantic composition, in contrast, enhances service descriptions with semantic annotations, ontologies, and domain-specific knowledge, enabling a more intelligent and meaning-aware composition process. This approach improves service discovery, selection, and integration by considering not only the structure but also the intended functionality of web services.

Several semantic description languages facilitate this approach, including:

- OWL-S – Supports capability-driven service descriptions based on ontologies.
- WSDL-S – Extends WSDL by adding semantic annotations for service capabilities.
- WSMO (Web Service Modeling Ontology) – Provides a framework for describing semantic web services.
- SAWSDL (Semantic Annotations for WSDL and XML Schema) – Introduces semantic metadata to WSDL, enhancing service interoperability.

By incorporating semantic reasoning, semantic composition enables more accurate service matching, automated service integration, and enhanced adaptability in dynamic environments.

3.5.6 Service Composition (Manual, Semi-automatic, or Automatic Composition)

Manual composition requires human developers or system architects to directly define and specify the composition logic [105]. This involves:

- Manually defining the control flow (abstract workflow) and data flow for the composition process.
- Using WS composition tools that support standard languages such as BPEL, or manually configuring web services.
- Explicitly connecting web services to operations within the abstract workflow.

This approach allows developers to precisely tailor the composition according to specific control and flexibility requirements. However, it has several drawbacks:

- Time-consuming and error-prone, requiring significant manual effort.
- Limited scalability, especially for large-scale or complex compositions.

Semi-automated composition provides tool-assisted support for the composition process while still requiring some level of user involvement [107]. These tools typically offer:

- Service discovery to help identify suitable services.
- Matching mechanisms based on functional or semantic similarities.
- Recommendations for service compositions using established patterns or best practices.

While users retain control over the composition, the tool accelerates the process and reduces errors. This approach is particularly useful for moderately complex compositions, where users benefit from guidance in selecting appropriate services. However, it is not yet fully scalable for large-scale service environments.

Automatic composition eliminates the need for direct user involvement in defining the composition process [105]. Instead, the system:

- Analyzes the service request, including functional requirements and constraints.
- Automatically generates an optimized composition plan, selecting and integrating suitable services.

This approach is particularly effective in dynamic and volatile environments, such as:

- Intelligent environments, where services need to be composed in real-time.
- Mobile and adaptive systems, where service availability fluctuates, and user needs change dynamically.

By leveraging automation, automatic composition enhances efficiency, adaptability, and scalability, making it ideal for personalized and rapidly evolving service ecosystems.

3.6 Approaches to Web Service Composition

Research on Web Service Composition (WSC) has gained significant attention, leading to the development of various composition methodologies. Existing approaches can be broadly classified into the following categories:

3.6.1 Automated WSC Approaches

The majority of the automated WSC approaches fall into one of these categories: workflow-based methods and AI-planning methods

3.6.1.1 Further Subsection

One of the earliest approaches to automatic Web Service Composition (WSC) is the workflow-based composition method. The primary objective of workflow-based planning is to automatically generate a sequence of web services that fulfill a specific user request or application requirement. This approach integrates concepts from automated planning and workflow modeling, ensuring a structured execution of services. In this method, the workflow acts as an action plan, defining:

- The sequence in which different web services (WS) are invoked.
- The dependencies between services by linking outputs from one service to inputs of another.
- The overall logical structure of the composite application [109].

Two widely used workflow-based composition techniques include:

- Workflow Net-Based Composition (Workflow Nets) [113] - Uses Petri nets to model and analyze service workflows.
  - Ensures correctness by verifying the soundness of service interactions.
  - Suitable for formal verification of service execution.

- Business Process Modeling Notation (BPMN) [114] - A graphical representation standard for modeling business processes and service workflows.
  - Provides intuitive visualization of service interactions.
  - Widely used in business process automation and enterprise service integration.

By leveraging workflow-based planning, WSC achieves structured and efficient service execution, making it an essential technique for automated service composition in various domains.

3.6.1.2 AI-planning Methods

In AI planning-based Web Service Composition (WSC), the composition problem is framed as a planning problem, where a sequence of actions is generated to transition from an initial state to a desired final state [115, 116]. The plan serves as a structured representation of how actions are organized in space and time to achieve the intended service composition. Many existing methods for automatic WSC rely on AI planning techniques, which can be categorized into several key approaches, including rule-based planning, Planning Domain Definition Language (PDDL), Hierarchical Task Network (HTN) planning, logic-based planning, and graph-based planning [109].

Rule-based planning enables flexible web service composition based on predefined rules and conditions, similar to if-then-else statements. This method allows developers to establish criteria that define how services should be combined, ensuring dynamic and adaptable compositions. However, as the number of rules and services grows, managing their interactions and dependencies becomes increasingly complex. This can make the composition logic difficult to understand, troubleshoot, and maintain, especially in large-scale systems. An example of this strategy is presented in [117].

To address the limitations of rule-based approaches, PDDL (Planning Domain Definition Language) was introduced to standardize the input data for planners [118]. PDDL provides a rich and expressive language that facilitates the description of planning problems, knowledge representation, actions, domain-specific goals, and constraints. By enabling precise specification, PDDL makes it easier to reason about service composition. However, its use requires a deep understanding of planning concepts and strict syntax, making it less accessible to non-expert users. Some researchers have proposed solutions that convert OWL-S descriptions into PDDL, considering both functional and non-functional properties in service composition [119].

Another widely used method is Hierarchical Task Network (HTN) planning, which decomposes complex tasks into smaller subtasks, resulting in a hierarchical structure [120]. This modularity simplifies service composition, making it easier to develop, maintain, and reuse. However, defining the hierarchical structure, decomposing tasks, and specifying methods and operators require careful design and a strong understanding of HTN planning concepts. As the composition size increases, the complexity of HTN-based planning may also rise, making it difficult to manage large-scale compositions efficiently. Researchers have proposed HTN-based automatic service composition methods that incorporate both functional and non-functional properties to enhance service integration [121].

In addition to HTN, logic-based planning offers another structured approach to web service composition. This method defines a set of composition rules using first-order logic, where constraints and logical dependencies form the basis of the composition plan. While logic-based planning ensures formal correctness, it faces challenges related to scalability, particularly when dealing with large-scale compositions with numerous interdependent tasks. Some studies, such as [122], have explored situational calculus-based logic planning, while others have proposed modifications to the Golog framework to incorporate user preferences in service composition [123].

Another approach, graph-based planning, represents service composition as a directed acyclic graph, where nodes correspond to web services and arcs represent semantic dependencies. The composition process is divided into two modules: one for creating the planning graph and another for identifying the optimal composition scheme. The solution search operates within a space consisting of states, actions, transaction functions, and cost functions. Two common strategies used in graph-based planning are forward search, which begins from the initial state and moves toward the goal state, and backward search, which starts from the goal and traces back to the starting state [109]. While this method provides a structured way to identify optimal compositions, the search space can grow rapidly, making computation challenging. To address this issue, heuristic functions and specific algorithms are often employed to estimate the usefulness of different search paths. Additionally, adapting graph-based models to dynamic changes in real-time environments can be difficult, as modifying the graph structure to accommodate new services, requirements, or constraints may impact the overall efficiency of the composition process.

Some researchers have extended graph-based planning by incorporating contextual information into web service descriptions. A method proposed in [124] suggests using an ontology-based context model to enrich service descriptions with additional contextual data. The planning graph is then built using a semantic and contextual service discovery process, where pruning techniques with backtracking are applied to refine the composition search. Finally, a classification step assigns scores to candidate solutions, helping determine the most suitable service composition based on defined criteria.

Overall, AI planning techniques provide a structured and automated approach to web service composition, offering flexibility, efficiency, and adaptability. While rule-based and logic-based methods ensure structured reasoning, HTN, PDDL, and graph-based planning facilitate task decomposition, workflow structuring, and search optimization. The choice of planning approach depends on system complexity, adaptability needs, and computational constraints, making it essential to balance automation, accuracy, and scalability in WSC.

3.6.2 QoS-aware Service Composition

In QoS-aware Web Service Composition (QWSC), the service selection process plays a crucial role in ensuring that the composed service meets specific Quality of Service (QoS) requirements. QoS in Web Services (WS) consists of a range of non-functional criteria, which help differentiate between services that provide the same functionality. Each WS operation is associated with a set of QoS attributes, which accurately reflect user needs and significantly influence user satisfaction. While Service-Oriented Architecture (SOA) design recognizes up to 13 key quality criteria [125], most research in QWSC prioritizes five main QoS criteria: response time, reputation, price, reliability, and availability [126]. These factors are critical in selecting the best-performing WS.

To ensure that QoS constraints are met at both local and global levels within a service composition, researchers from various domains, including Cloud Computing [127, 128] and IoT [129–131], have focused on enhancing QWSC techniques. Unlike automatic service composition, which primarily deals with service discovery and integration, the objective of QWSC is to select the optimal service from a set of candidate services for each task, ensuring that the final composition adheres to QoS constraints [115].

A promising direction for solving QWSC challenges lies in Nature-Inspired Computing (NIC) approaches [132]. NIC is an innovative discipline that develops new computational techniques by drawing inspiration from various biological and natural systems, including evolution, ecology, development, and behavior [164]. This has led to groundbreaking research in areas such as neural networks, swarm intelligence, evolutionary computing, artificial immune systems, and multi-agent systems. These bio-inspired methodologies have been successfully applied to optimization problems, machine learning, data mining, and fault diagnosis. Given their ability to efficiently handle complex, large-scale, and dynamic optimization problems, nature-inspired algorithms offer promising solutions for QoS-aware service composition, enabling adaptive, scalable, and high-performance WS selection.

The integration of Nature-Inspired Computing (NIC) in QoS-aware Web Service Composition (QWSC) is driven by several key challenges inherent to the WS selection and composition process [132]. These challenges include computational complexity, diverse user preferences, and the dynamic nature of web services, which necessitate adaptive and efficient optimization techniques. The main motivations for employing this paradigm in QWSC can be summarized as follows:

First, the WS selection and composition problem is classified as an NP-hard optimization problem, meaning that finding an optimal solution requires exponential time [134]. Due to this computational complexity, Meta-heuristic optimization algorithms offer a practical alternative by providing near-optimal solutions within a reasonable time. These algorithms, inspired by natural processes such as evolution and swarm intelligence, enable efficient exploration of large search spaces, making them well-suited for addressing the computational challenges of QWSC.

Second, multi-objective optimization is essential in QWSC because users have diverse preferences, and opinions may vary across individuals. Traditional single-objective approaches may fail to accommodate different user requirements, whereas multi-objective meta-heuristic methods [130] can generate a set of optimal solutions, allowing users to select the one that best aligns with their specific needs. This flexibility enhances user satisfaction and makes the service selection process more personalized and adaptable.

Finally, the dynamic nature of web services introduces additional complexity. WS are deployed and executed over the internet, where service availability and QoS parameters can change unexpectedly. For instance, a previously selected service component may suddenly become unavailable, or QoS values may fluctuate due to network conditions or system load. In such cases, users may need to modify their choices and constraints in real time. Traditional planning methods often require replanning the entire composition, leading to delays and inefficiencies. However, reinforcement learning techniques [147], which continuously maintain a solution population, offer a more adaptive approach. These techniques can immediately generate a new composition schema whenever a change occurs, ensuring seamless adaptation and continuous optimization.

By addressing these challenges, nature-inspired meta-heuristic approaches provide robust, scalable, and adaptive solutions for QoS-aware service composition, making them an effective paradigm for managing the complexity and dynamism of modern web service environments.

3.6.2.1 Meta-heuristic Algorithms for QWSC

The problem of finding a complete composition of services that satisfies all imposed constraints is considered NP-hard due to the large number of available web services with similar functionalities and the need to satisfy a complex set of user constraints. As a result, solving this problem requires exponential time and computational cost.

Early research addressed the QWSC problem using linear programming techniques [148, 149], where the goal was to maximize a utility function that represents a weighted sum of user-defined QoS criteria. While these methods provided a structured approach, they often struggled with scalability and computational efficiency. More recent advancements have led to the formalization of QWSC as an optimization problem, solved through meta-heuristic algorithms [132, 134]. These algorithms efficiently explore the large search space to find satisfactory solutions within a reasonable time frame.

Among the different meta-heuristic approaches, single-solution-based methods, such as Tabu Search [150] and Simulated Annealing [151], have been applied to solve the QWSC problem. However, the majority of solutions leverage population-based Meta-heuristics, which are more effective for handling multi-objective optimization problems. These include Genetic Algorithms (GA) [130, 152], Particle Swarm Optimization (PSO) [153], Ant Colony Optimization (ACO) [154], Artificial Bee Colony Algorithm (ABC) [155, 156], and Harmony Search Algorithm (HSA) [115].

By employing these meta-heuristic techniques, the QWSC problem can be formulated as a multi-objective optimization problem [130, 156], allowing for the selection of the best composition based on multiple QoS criteria. These methods provide scalable, adaptable, and efficient solutions, making them well-suited for the complex and dynamic nature of modern web service environments.

3.6.2.2 Reinforcement learning for QWSC

To enhance adaptability in WSC, recent research has focused on Reinforcement Learning (RL) as a powerful paradigm for adaptive decision-making in QoS-aware Web Service

Composition (QWSC). RL enables intelligent agents to learn and refine their decision-making policies based on environmental feedback, making it particularly useful in uncertain and dynamic environments where service performance, availability, or other key factors may fluctuate over time [157].

Several RL-based approaches have been proposed to address the QWSC problem. In [157], multi-task deep reinforcement learning was applied to facilitate IoT service selection in smart city environments, optimizing decision-making across multiple QoS criteria. For handling large-scale dynamic service composition, [158] introduced the Adaptive Deep Q-learning and RNN Composition Network (ADQRCN), which effectively adapts to changing service conditions. Similarly, [159] employed Markov Decision Process (MDP) models to implement an adaptive service composition strategy.

To further enhance the efficiency of service composition, [160] proposed a hierarchical reinforcement learning algorithm, enabling more structured and scalable decision-making. Additionally, [161] combined Recurrent Neural Networks (RNNs) with RL, where QoS predictions were performed using an RNN, followed by dynamic service selection through reinforcement learning techniques.

These RL-driven approaches demonstrate significant potential in optimizing service composition, ensuring flexibility, efficiency, and adaptability in response to dynamic changes in service availability and QoS variations.

3.6.3 Service Compositions in Agent Technology

Agent technology offers a decentralized and distributed approach to Web Service Composition (WSC), enhancing the scalability and flexibility of service-based systems. By leveraging autonomous agents that can communicate, collaborate, and negotiate, agent-based approaches enable the dynamic selection and integration of web services, making them particularly effective in complex and dynamic environments.

A notable example is the Belief-Desire-Intention (BDI) agent model [162], a deliberative agent architecture designed to improve the intelligence of semantic web service composition. BDI agents operate by reasoning about their beliefs (knowledge of the environment), desires (goals to achieve), and intentions (current plan of action), allowing for more context-aware and adaptive service compositions.

The use of multi-agent systems (MAS) further enhances WSC by enabling distributed coordination among multiple agents. In MAS-based service composition, agents handle service discovery, selection, creation, and execution while considering semantic interoperability between service parameters. A multi-agent architecture that supports dynamic composition and end-to-end quality assurance was introduced in [112], demonstrating the effectiveness of agent-based techniques in optimizing service workflows.

Additionally, recent research has explored the integration of MAS with microservices. In [163], a novel framework called Multi-Agent Microservices (MAMS) was proposed, combining the benefits of MAS-driven service orchestration with microservices architecture for dynamic semantic web service composition. This approach enhances scalability, modularity, and adaptability, making it a promising direction for next-generation service-oriented computing.

Multi-Agent Systems (MAS) offer several advantages for Web Service Composition (WSC) by enhancing flexibility, scalability, communication, and intelligence. Their decentralized and distributed nature makes them highly effective in dynamic and complex service environments.

One of the key benefits of MAS-based WSC is improved flexibility and adaptability. Unlike centralized systems, MAS are designed with more adaptable structures, where each agent is responsible for a specific task or service. These agents can dynamically adjust to environmental changes, modifying their behavior accordingly to ensure smooth service composition and execution.

Another advantage is decentralization and distribution, which allows MAS to distribute the workload among multiple agents. This improves performance, scalability, and fault tolerance, as service composition tasks are not reliant on a single point of failure. By decentralizing decision-making, MAS reduces bottlenecks and enhances the resilience of service-oriented systems.

Enhanced communication and coordination is another major strength of MAS. Agents can collaborate and share information to achieve a common goal, ensuring seamless problem-solving and exception handling. This results in a more efficient and effective service composition process, minimizing disruptions and optimizing workflow execution.

Finally, MAS increase intelligence and autonomy in WSC. Agents can be designed with decision-making capabilities, allowing them to operate independently without requiring constant human intervention. This autonomy enhances the efficiency of service composition, enabling self-adaptive and intelligent decision-making in response to changing service conditions.

By leveraging these benefits, MAS-based approaches provide a robust and scalable framework for dynamic and distributed WSC, making them well-suited for modern service-oriented computing environments.

3.6.4 Hybrid Approaches

Hybrid approaches integrate elements of AI planning, graph-based techniques, and optimization methods to solve the WSC problem while ensuring both functional correctness and optimized QoS solutions [115, 165, 166]. By combining the strengths of different methodologies, hybridization offers a balanced and efficient approach to handling complex service composition scenarios.

One of the primary advantages of hybrid approaches is their ability to handle complex objectives and constraints. WSC often involves multi-objective optimization, where multiple constraints must be satisfied while optimizing performance metrics such as response time, reliability, and cost. Hybridization addresses this challenge by leveraging planning techniques to explore the most promising areas of the search space, while Meta-heuristics refine the search to identify the best possible solution. By combining structured search with heuristic optimization, hybrid methods effectively balance feasibility and optimality in service composition.

Another key benefit is adaptability in highly dynamic environments. The service landscape constantly evolves as services are added, removed, or modified, making it necessary to adapt compositions in real time. Hybrid approaches use planning techniques to generate a high-level solution, which is then further optimized and adjusted using Meta-heuristics. This ensures that service compositions remain robust and efficient, even as environmental conditions change.

Additionally, hybridization enhances interpretability and transparency in the composition process. While meta-heuristic optimization can sometimes be viewed as a black-box approach, planning techniques provide a more structured and explainable solution. By first generating a high-level composition strategy through planning and then refining it through optimization, hybrid methods offer better insight into both the composition structure and the optimization process.

By integrating AI planning, graph-based modeling, and meta-heuristic optimization, hybrid approaches offer a powerful and scalable solution for QoS-aware WSC, ensuring functional correctness, dynamic adaptability, and optimized performance.

3.7 Key Issues and Challenges in WSC

Despite extensive research in WSC, several unresolved challenges remain. The rapid advancement and adoption of emerging computing paradigms such as cloud computing, social computing, and the Internet of Things (IoT) have further complicated the composition process [167]. These developments introduce new complexities, constraints, and dynamic requirements, making efficient service composition increasingly challenging. The following issues represent some of the key challenges in WSC that require further attention:

3.7.1 Web Service Composition Validation and Evaluation

Errors are inevitable in Web Service (WS) design, and as with other distributed systems, managing the behavior of composite services can be highly challenging. It is crucial to detect and resolve design flaws before the actual deployment of composite services, as runtime errors can lead to unexpected behaviors and significant losses [168].

A major limitation in most service composition methods is the lack of built-in mechanisms for verifying compositional correctness [167]. This verification process involves two levels of correctness:

- Basic Correctness: Ensures that the execution of the composite service follows a valid control flow, preventing issues such as deadlocks that could cause execution failures.

- Property-Based Correctness: Validates that the composite service satisfies specific functional and non-functional requirements, including safety properties (ensuring no harmful behaviors occur) and liveness properties (ensuring progress toward the desired outcome).

The most effective way to ensure correctness in WSC is to integrate compositional correctness verification directly into the composition algorithm itself [107]. By embedding formal verification techniques, such as model checking, theorem proving, or runtime monitoring, WSC methods can ensure that the generated compositions meet correctness constraints, reducing the risk of failures after deployment.

3.7.2 Composition of Web services in a Dynamic, Distributed, and Automatic Manner

Web Service (WS) environments are inherently dynamic, with services being added, removed, or temporarily unavailable at any given moment. Additionally, the capabilities and QoS attributes of existing services may change over time, making it challenging to ensure continuous and reliable service composition [107].

To address these challenges, dynamic service composition systems must support real-time service adaptation by enabling the automatic detection and replacement of component services with alternatives that offer better quality or improved functionality as needed. This adaptability ensures that service compositions remain robust and high-performing, even in fluctuating environments.

Moreover, the business landscape itself is highly dynamic, requiring service-oriented systems to rapidly respond to evolving requirements. To meet these demands, promising research is focusing on self-managing service composition frameworks with capabilities such as self-formation, self-improvement, self-healing, and self-adaptation [168–170]. These autonomous features enable services to continuously learn, optimize, and adjust their behavior in response to environmental changes, reducing the need for manual intervention and enhancing the overall resilience and efficiency of WSC.

3.7.3 Composition of Services in a Pervasive Environment

The proliferation of ubiquitous and interconnected computing devices has enabled mobile users to seamlessly access WS [131, 167, 170, 171]. However, designing and composing services for a wide range of mobile devices introduces new challenges, including the need to address context awareness, device heterogeneity, and user personalization requirements.

Context awareness is essential in mobile environments, where factors such as location, network conditions, battery life, and device capabilities can impact service execution. Additionally, device heterogeneity, where different mobile devices have varying processing power, screen sizes, and operating systems—complicates service adaptation and interoperability. Furthermore, user personalization demands that composite services be tailored to individual user preferences and dynamic needs.

To ensure efficient and reliable composite service execution, special considerations must be made regarding the resource constraints inherent in pervasive and mobile environments. Composite services must be optimized for low bandwidth, intermittent connectivity, and limited processing power, ensuring that service quality and user experience remain uncompromised despite the challenges posed by mobile computing.

3.7.4 Semantic WSC Using Ontologies

Ontology serves as an explicit specification of conceptualization, defining terms and their relationships in a way that is understandable to both humans and computers. In open and dynamic environments, different ontologies are developed to enable knowledge sharing across various domains. However, service providers and requesters may interpret shared knowledge differently, leading to potential misalignments in service descriptions and requests [92, 173].

This issue becomes particularly significant when different ontologies are used to annotate the same services and requests, resulting in semantic discrepancies that hinder effective service discovery, matching, and integration. To address this challenge, ontology mapping techniques are required to ensure coordination and interoperability between diverse ontologies [167]. These techniques facilitate the alignment of concepts, relationships, and terminologies, allowing heterogeneous systems to seamlessly interact and ensuring that service compositions remain consistent and meaningful across different knowledge representations.

3.7.5 Service Governance

E-Government aims to provide efficient and accessible government services to citizens, businesses, and stakeholders by leveraging the power of the Internet and Information and Communication Technologies (ICTs) [166]. However, the implementation of E-Government systems presents significant challenges, particularly in ensuring interoperability, integration, and security among various government services and platforms.

Due to the complex and heterogeneous nature of government services, WSC plays a crucial role in enabling seamless service integration across different departments and agencies. The dynamic composition of WS in E-Government is of particular interest to researchers, as it allows for adaptive and scalable service architectures that can respond to changing regulatory, technological, and societal needs. Additionally, semantic web technologies enhance the structuring and interpretation of composite services, ensuring that E-Government WSC is not only functionally efficient but also semantically meaningful [165].

By addressing these challenges, WSC in E-Government can facilitate automated service delivery, improved transparency, and enhanced citizen engagement, ultimately contributing to the modernization and efficiency of public administration.

3.8 Conclusion

Integrating semantic technologies and web service composition represents a paradigm shift in distributed systems. While semantics enhance interoperability and automation, composition improves modularity and scalability. Together, they address critical challenges and pave the way for more intelligent, adaptive, and efficient systems. Continued research and technological advancements will further solidify their role in e-government, healthcare, and IoT domains. Based on this chapter, the next chapter will introduce a conceptual model for our work in this thesis, providing a structured approach to implementing these technologies effectively.

Chapter 4

Conceptual Model and contributions

4.1 Introduction

As Algeria continues its digital transformation, developing a robust and efficient e-government system remains a national priority. However, one of the most pressing challenges that undermines this progress is the lack of interoperability between fragmented public sector information systems. This fragmentation hinders effective communication, impedes seamless data exchange, and disrupts the integration of essential public services. In response to these limitations, this chapter explores innovative solutions based on Semantic Web technologies that are capable of structuring, linking, and semantically enriching information to promote a more unified and collaborative digital governance system. This chapter presents three key contributions that collectively aim to address these challenges. The first part presents a conceptual model for citizen request processing, designed to manage and meet user needs through a structured semantic approach. The second part details the development of a citizen profile ontology, which plays a key role in enhancing the personalization and accessibility of e-government services by enabling consistent and compliant representation of citizen data. Finally, the third part focuses on the problem of Quality of Service (QoS)-aware service composition, proposing a Discrete Balanced Teaching-Learning-Based (DBTLBO) algorithm for dynamically selecting and configuring optimal services based on performance criteria.

4.2 Part 1: Conceptual Model for Citizen Request Processing System

In recent years, the growing interest in citizen adoption of e-government has spurred significant research efforts, resulting in the development of numerous projects aimed at enhancing public service delivery through advanced architectural strategies. The integration of semantic technologies into e-government frameworks is a major area of interest in this field. These technologies enhance the descriptions of services provided by various public administrations by adding semantic information, enabling automated service processing and effective information exchange among stakeholders. This leads to more coherent, interpretable interactions and fosters improved integration and collaboration across services. Ontologies and Semantic Web Services (SWS) are two of the most widely used technologies in this context.

In the following, we first present a set of projects using ontologies and supporting services (SWS) to develop an e-government infrastructure, and then describe our conceptual model that leverages ontologies to efficiently classify, process, and satisfy citizen requests.

4.2.1 E-government Projects

Several pioneering e-government projects illustrate how Semantic Web Services (SWS) and ontologies have been applied to improve the delivery, integration, and interoperability of public services. Although these initiatives pursue similar objectives, they approach semantic technologies from different angles, leading to the development of varied frameworks, models, and tools that address challenges such as data representation, service composition, and administrative process management. The table 4.1presents a comparison between these initiatives based on four key dimensions: “Interoperability and Ontologies”, “SWS Model”, “SWS Architecture” and “User-Oriented Service Portal”.

The Access-eGov project (AeG, 2006-2010) [176] developed a functional architecture based on Semantic Web Services (SWS) technology to address the challenges of administrative process integration and interoperability. This objective was achieved through the creation of a comprehensive set of ontologies. In addition, the project introduced tools for creating service access portals, facilitating user interaction in the e-government field. To support the description of services, a specialized markup language, GovML, was developed [177], which defines a set of metadata for representing public administration services and life events.

The SemanticGov project (2006–2009) focused on addressing service integration and interoperability at both national and municipal levels. At the national level, the project introduced the National European e-Government Services (NEGS), which facilitates interaction among public administrations within the same country. At the municipal or cross-border level, it established the Pan-European e-Government Services (PEGS), aimed at enabling cooperation between public administrations from different European countries. PEGS emerged from the challenge of integrating diverse NEGS frameworks. Both levels share the common goal of resolving inconsistencies such as variations in service names, document formats, and the use of administrative and legal terminology. To meet the interoperability requirements of these systems, SemanticGov developed an infrastructure for the provision of Semantic Web Services (SWS), supporting seamless collaboration both within and across national borders [178].

The TerreGov project (2004–2007) focused on the modeling of "administrative procedures" from the perspective of public administrations (PAs), specifically targeting civil servants as its main users. Its objective was to streamline and enhance the internal workflows of PAs by enabling the design, execution, and coordination of administrative processes in a more structured and interoperable manner. To achieve this, TerreGov adopted widely accepted W3C standards, ensuring compatibility and extensibility across different systems and administrative contexts. It utilized web services technology for the implementation of modular and reusable services offered by individual PA agencies. For managing the execution and coordination of these services, the project integrated a workflow management system based on Business Process Execution Language (BPEL), which allowed for the composition and orchestration of complex administrative tasks. Additionally, TerreGov incorporated a semantic access layer using the OWL-S ontology, enabling more intelligent service discovery, automated reasoning, and meaningful interaction between users and services [179].

The DIP project (2004) was a pioneering initiative in building an infrastructure based on Semantic Web Services (SWS). It introduced an e-government ontology and a reference model that promotes autonomy among public administrations (PAs), allowing them to define their own concepts and use different terms for the same ideas without necessarily sharing a unified perspective. To enable interoperability despite these differences, DIP employed a standardization process using high-level ontologies such as D&S (Descriptions & Situations) from the DOLCE framework. The domain ontology was developed with input from domain experts and built upon specialized ontologies.

The development of SWS within DIP is grounded in three types of ontologies defined in its reference model, alongside the e-government ontology. This ontology enables the semantic description of various components of public service systems and is entirely based on the Web Service Modeling Ontology (WSMO) and the Operational Conceptual Modeling Language (OCML) [180]. DIP integrates all core WSMO components—Ontology, Goals, Web Services, and Mediators—for modeling e-government services. Web Services represent the functional aspects of public services from the perspective of the PA, while the Goal ontology captures the needs and expectations of end users, such as citizens and businesses, serving as a semantic bridge between services and life event ontologies. Mediators address interoperability challenges.

The OntoGov project (2003–2006) addressed the pressing challenge of change management in e-government systems by developing the first comprehensive framework for reconfigurable services, capable of adapting to legislative changes, policy updates, and organizational restructuring—common factors that disrupt public service delivery. To support this adaptability, OntoGov employed semantic technologies to enhance the configuration and reconfiguration of e-government services, creating a suite of systems that manage the entire service lifecycle, from modeling to discovery [181]. The OntoGov platform includes tools for ontology development, service description, configuration/reconfiguration, and deployment, all built on W3C standards. It makes use of Semantic Web (SW) technologies, along with OWL and KAON languages [182], to ensure semantic consistency and interoperability across services.

TABLE 4.1: Comparison of SWS-Based E-Government Projects

Project	**Interoperability and Ontologies**	**SWS Model**	**SWS-Based Architecture**	**User-Oriented Service Portal**
Access-eGov (AeG)	Developed comprehensive ontologies and GovML metadata for semantic interoperability.	Services described using GovML; focuses on metadata for life-event services.	Functional architecture based on SWS to support public administration integration.	Includes tools for service portals; user access modeled on life events.
Semantic-Gov	Ontologies for NEGS/PEGS to address national and cross-border interoperability.	SWS infrastructure for service description and discovery.	Infrastructure for seamless collaboration between administrations.	Not explicitly user-focused; oriented toward backend interoperability.
Terre-Gov	Semantic access layer using OWL-S; adopts W3C standards.	OWL-S enables intelligent discovery and automated reasoning.	Architecture based on web services and BPEL workflow orchestration.	Designed mainly for civil servants; limited focus on public-facing portals.
DIP	Uses high-level ontologies (D&S, DOLCE); allows diverse vocabularies with alignment.	E-government ontology with flexibility for PA-defined concepts.	SWS infrastructure promoting autonomous service description across PAs.	Focuses on semantic mediation, not explicitly user-facing.
Onto-Gov	Employs OWL and KAON; ensures semantic consistency across services.	Semantic technologies for configuration and reconfiguration of services.	Reconfigurable architecture managing the full service lifecycle.	Supports service adaptability but no explicit mention of user-facing portals.

4.2.2 System Overview

The proposed system provides a structured workflow for handling citizen requests using ontology-based classification and service composition. The model consists of several key components that collaborate to process requests effectively and assign them to the appropriate governmental department or institution, ensuring efficiency and accuracy in service delivery (Figure 4.1).

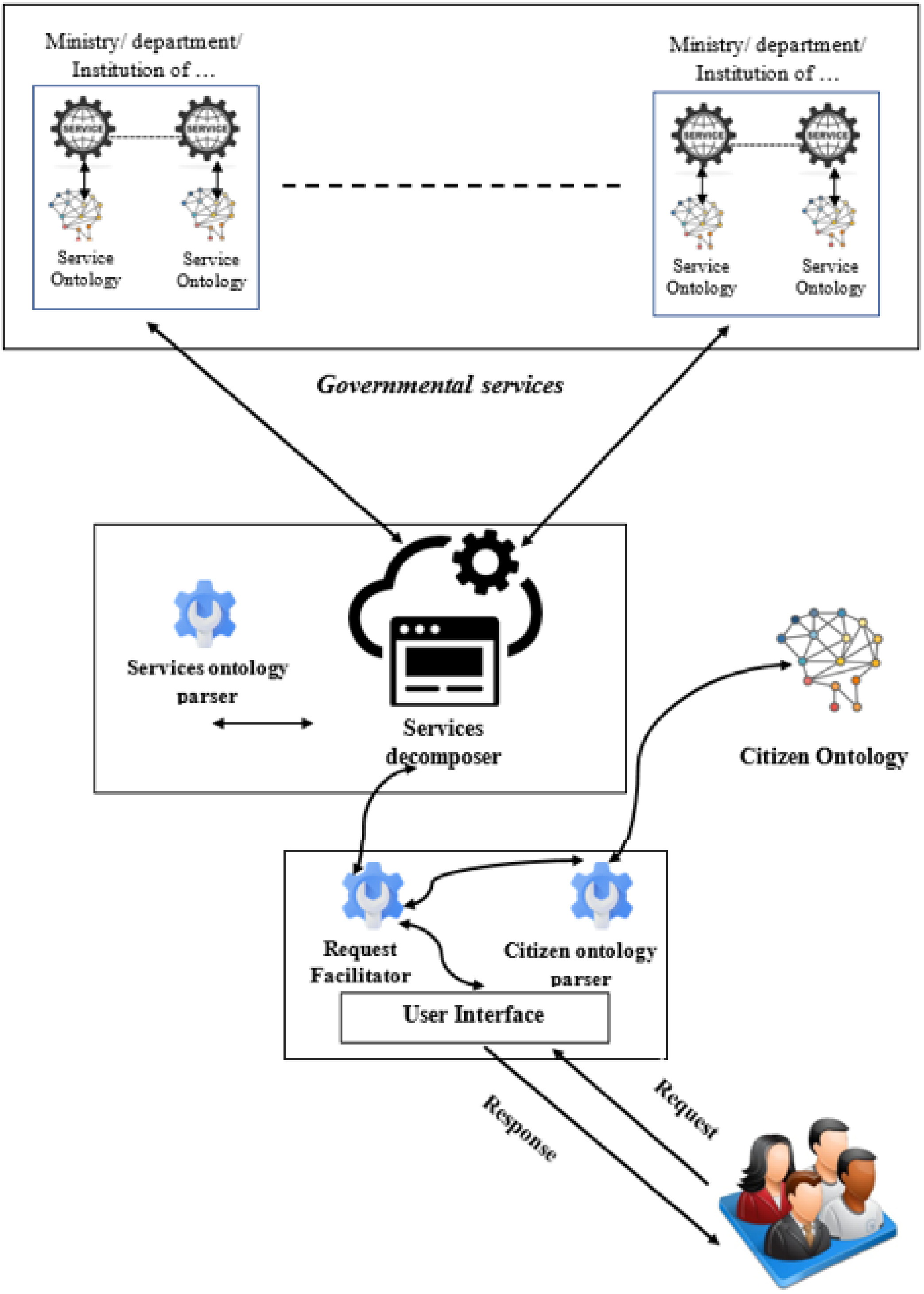

FIGURE 4.1: Conceptual model for Citizen Request Processing System

The citizen's request will be passed through four steps to get the data needed, which are:

- Citizen Request Submission
- Reformulate and enrich the query
- Service Composition and Processing.
- Response Handling and Citizen Feedback

4.2.2.1 Citizen Request Submission

The process begins when a citizen submits a request via the user interface, which is the main communication link between users and the system. Acting as a graphical user interface (GUI), it provides access to front-office applications that allow citizens and businesses to express their needs, enter relevant data, and search for or use available e-government services. This interface allows users to express their requests in a structured or semi-structured manner. Designed to be accessible and easy to use, it allows people from all backgrounds to navigate and interact effectively with the platform.

When a citizen submits a request via this interface, the request is forwarded to the Request Facilitator, a core module responsible for managing the initial processing. The request facilitator interacts with the Citizen Ontology Parser to retrieve relevant details about the citizen, including their profile, activity type, and other contextual information. This classification helps determine the best approach for handling the request.

4.2.2.2 Reformulate and Enrich the Query

Once the request is submitted, it is directed to the Request Facilitator, a central module responsible for managing its initial processing. The query is then executed using SPARQL (SPARQL Protocol and RDF Query Language) [184] on the mediator's global ontology. SPARQL serves both as a protocol and a query language designed to access RDF data. Its primary role is to interpret application requirements, expressed as queries, and return the results in the form of RDF graphs. Moreover, SPARQL allows for querying diverse data sources and retrieving relevant information without requiring prior knowledge of the data's structure or content [183]. As part of its operation, the Request Facilitator consults the Citizen Ontology Parser to extract relevant details about the citizen, such as their profile, type of activity, and contextual information. This classification guides the system in selecting the most appropriate method for processing the request.

The request facilitator also:

- Filters duplicate or similar requests to prevent redundancy.
- Prioritizes requests based on urgency and predefined government service levels.
- Enforces security and privacy policies to safeguard citizen data.
- Integrates with national identity systems to verify user credentials and eligibility.

The Citizen Ontology Parser utilizes a pre-defined Citizen Ontology database to extract additional insights about the citizen. The query will thus be semantically enriched with terms used in the Citizen Ontology.

By leveraging structured knowledge, this component classifies the citizen based on predefined attributes such as: Civil, Education, Job, Health, Document, and Properties.

The extracted information is then used by the request facilitator to refine and remodel the original request to ensure accuracy in processing.

4.2.2.3 Service Composition and Processing

After refining the request, it is passed on to the Services Composer, which is responsible for determining the most appropriate response by selecting relevant services.

The Services Composer contains a composition's algorithm that will find the best web service composition based on the enriched query and relies on the Services Ontology Parser, a module that interprets service-related ontologies. This parser identifies the nature of the request and matches it with available governmental services. Key functionalities include:

- Decomposing complex requests into multiple service components.
- Mapping requests to available services within governmental institutions.
- Identifying service dependencies and relationships to determine service hierarchy.
- Suggesting alternative solutions if the requested service is unavailable or has eligibility restrictions.

The services ontology helps establish logical connections between web services by analyzing metadata descriptors. This approach ensures efficient request decomposition and routing.

The remodeled request, now enriched with relevant classifications, is forwarded to the appropriate governmental department, ministry, or institution for processing. The system ensures that each request is directed to the correct service provider, reducing bureaucratic inefficiencies and enhancing service delivery speed. This step involves:

- Automated workflow generation, ensuring proper delegation to the right authority.
- Progress tracking for citizens to check the status of their requests.
- Escalation mechanisms if responses are delayed beyond standard service level agreements.
- Secure digital communication between departments to ensure rapid data exchange.

4.2.2.4 Response Handling and Citizen Feedback

Once the request is processed by the designated institution, the response is generated and communicated back to the citizen via the user interface. The system also integrates a feedback mechanism, allowing citizens to rate the service received and provide suggestions for improvement.

4.2.3 Part 1 Conclusion

This part has highlighted the crucial role that semantic technologies play in the development of e-government systems, particularly in improving the delivery of public services. Through a review of existing projects and the introduction of our own conceptual model, we demonstrated how the integration of ontologies and (SWS) contributes to more coherent, automated, and interoperable digital governance. Our proposed model enables the classification and processing of citizen requests with greater accuracy, significantly reducing processing time and improving user satisfaction. Furthermore, the model supports the provision of scalable services and enhances collaboration between entities through seamless data exchange. These features make the model a flexible and forward-looking solution capable of evolving to meet the growing demands of public administration. Overall, the approach presented in this chapter lays a solid foundation for developing resilient, citizen-centric e-government platforms.

4.3 Part 2: Citizens' Profiles Ontology for Improving E-government Services in Algeria

The electronic government environment of Algeria is characterized by disparate information systems and was created independently by different public ministries. This fragmentation often leads to a standardized lack of knowledge, posing major challenges for data interoperability and system integration. Semantic interoperability appears as a key solution, allowing more efficient and relevant exchange of information, ensuring that the data shared between the systems is semantically and contextually coherent.

The realization of semantic interoperability requires the development of a shared ontology that enriches data and services while recording and organizing shared semantic knowledge. Such an ontology promotes a coherent interpretation among various administrative entities, thus improving the efficiency, coordination, and overall quality of the provision of public services. However, designing a comprehensive ontology is a difficult task that requires methodological accuracy, domain knowledge, and the strategic application of semantic technologies.

4.3.1 The Development of Ontologies

This section focuses on the design and development of a citizen profile ontology adapted to the Algerian context. The proposed ontology aims to promote semantic interoperability and improve the accessibility, integration, and personalization of e-government services for Algerian citizens.

There are several definitions of ontology in the literature; the most commonly used definition is taken from Gruber [185]. He defined an ontology as an explicit specification of a conceptualization. A conceptualization refers to an abstract and simplified view of a domain of knowledge one wishes to represent for a certain purpose. The domain could be explicitly and formally represented using existing objects, concepts, entities, and the relationships that exist between them [185]. The domain could refer to a domain such as e-government.

In recent years, the development of ontologies has experienced considerable growth in the field of e-government thanks to their ability to effectively describe and structure government services offered to citizens. By facilitating the matching, mapping, and integration of various e-government services, ontologies play a crucial role in semantic interoperability and seamless service integration. The main motivations for adopting ontologies in e-government are [186]:

- Establishing a common understanding of structures among stakeholders;
- Enabling the extraction and processing of information from diverse sources and documents;
- Promoting the reuse of existing domain knowledge, thus saving time and resources;
- Providing a clear contextual framework for clearly communicating complex concepts;
- Offering a declarative representation of semantic content, independent of data format;
- Enable the comparison and integration of objects from heterogeneous information repositories.

Since the early 1990s, considerable research has been devoted to ontology design and construction [187–189]. Several methodologies have been proposed for developing ontologies, differing in the steps and tasks they suggest be performed during their construction. Despite these efforts, there is still no universally accepted standard methodology for ontology development [190]. However, as mentioned in [189, 190], METHONTOLOGY [101] is widely recognized as one of the most comprehensive and accomplished methodologies available. It clearly describes the key steps of ontology development through its seven defined phases: specification, knowledge acquisition, conceptualization, integration, formalization, implementation, and maintenance. Collectively, these phases form a structured life cycle for ontology development and comply with the criteria of the IEEE standard, ensuring a systematic and rigorous process.

The goal of the specification phase is to specify the ontology's main goal, the purpose for which it will be used, and the information that it will encompass. In the conceptualization phase, the acquired knowledge is arranged and organized through the construction of a controlled vocabulary, using external representations, which do not depend on languages or implementation environments. After the establishment of hierarchical relationships between the concepts and the construction of the taxonomy is made. Formalization and implementation phases concerned the conceptual model transformation to an implemented model, by using the Semantic web technology (ontology languages, standards, editors, and repositories). More details about these phases can be found in [101].

4.3.2 Citizen's Ontology

To create an ontology that encompasses the essential information required across various government departments undergoing digital transformation—covering personal, educational, health, and professional aspects—we adopted the METHONTOLOGY approach. For ontology construction, we utilized OWL (Web Ontology Language) due to its robust semantic capabilities and logical expressiveness. Additionally, we implemented the ontology using Protégé 5.5.0, a widely used, platform-independent tool that aligns with METHONTOLOGY standards. To ensure accuracy and consistency, we employed Pellet reasoner for ontology verification.

By systematically analyzing the personal, educational, professional, and health-related data integrated into Algerian e-government services, we developed a citizen ontology comprising 54 classes, 31 object properties, and 48 data properties.

4.3.2.1 Classes

The proposed citizen ontology is structured into several key classes: Civil, Education, Job, Health, Document, and Properties (Figure 4.2). Each class represents a distinct aspect of a citizen's profile, ensuring comprehensive coverage of personal and administrative information relevant to e-government services.

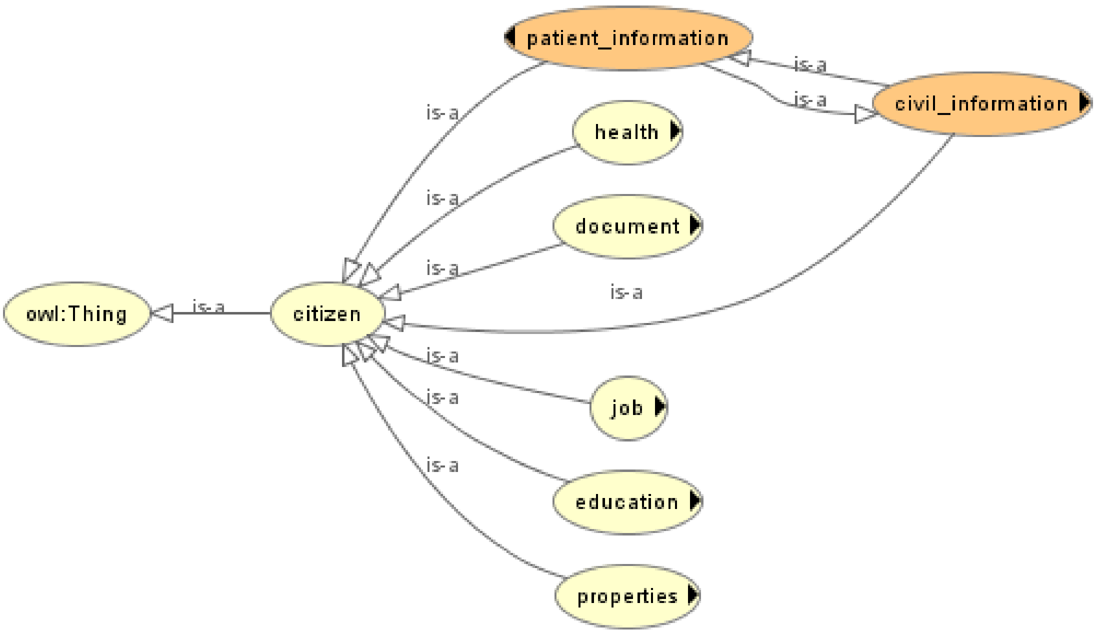

FIGURE 4.2: The citizen ontology

The Civil Information class (Figure 4.3) encompasses essential personal details about citizens, such as their name, parents' names, birthdate, place of birth, gender, address, marital status, blood type, and nationality. This class serves as the foundation for identifying individuals and linking them to various government services.

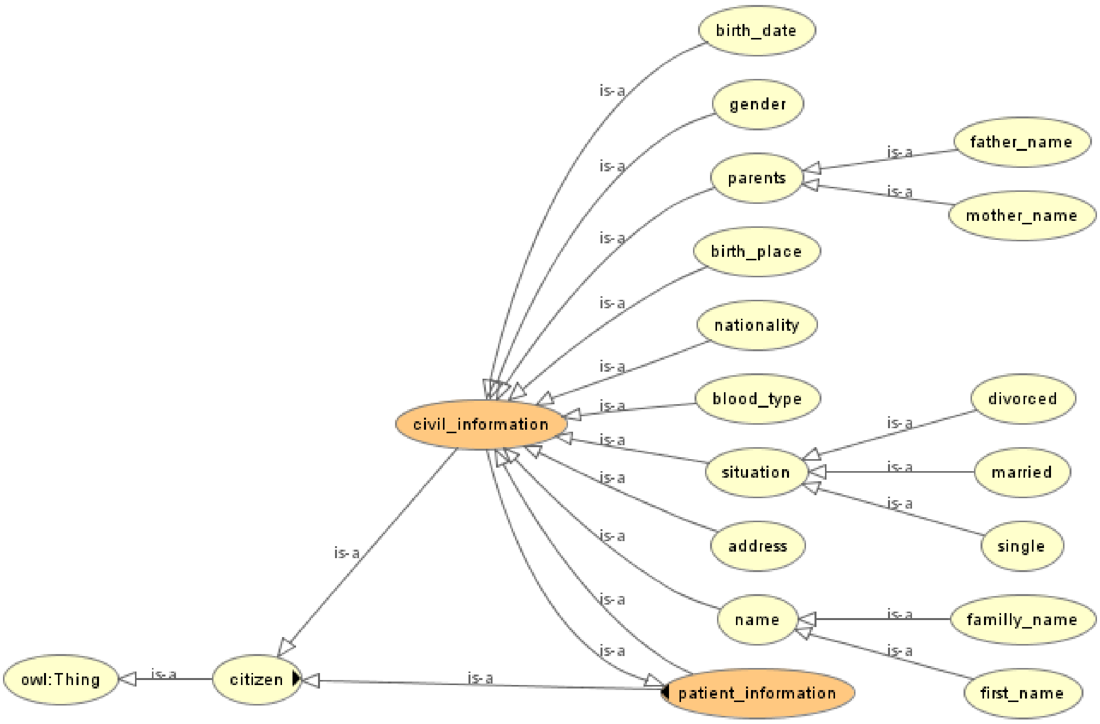

FIGURE 4.3: The civil information class ontology

The Education Level class (Figure 4.4) categorizes citizens based on their academic background. It is divided into two sub-classes: Not-Student and Student. The Not-Student sub-class includes illiterate individuals, while the Student sub-class represents those who have completed different stages of education, including primary school, middle school, high school, university, and vocational training. This classification enables the government to tailor educational and professional development programs to citizens' specific needs.

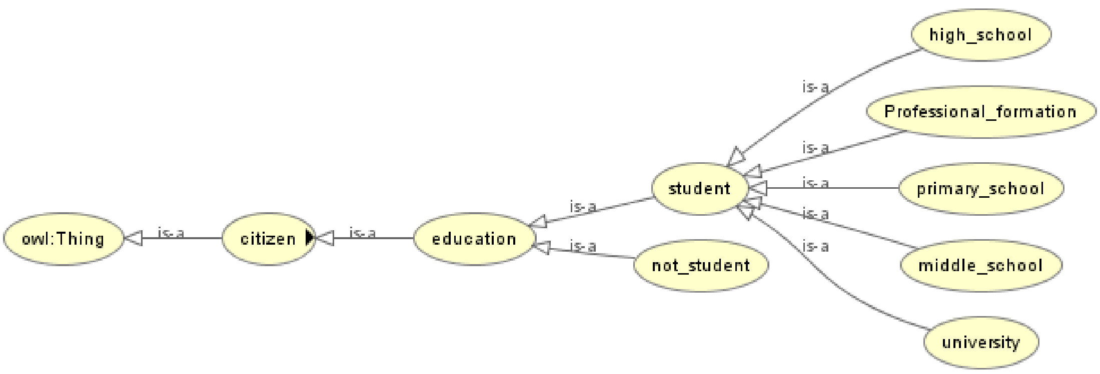

FIGURE 4.4: The education level class ontology

The Job class (Figure 4.5) represents a citizen's current employment status and is divided into two main sub-classes: Employed and Unemployed.

The Employed subclass identifies whether a citizen works in the public sector (government employment) or the private sector. This classification helps in organizing workforce data and facilitating employment-related services.

The Unemployed sub-class includes individuals who are currently not employed, allowing the system to track job seekers and provide targeted support, such as employment assistance programs or social benefits.

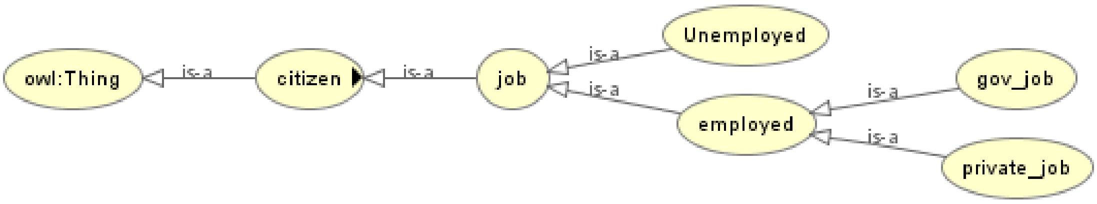

FIGURE 4.5: The job class ontology

The Health class (Figure 4.6) represents a citizen's health status and is categorized into two sub-classes: Healthy and Unhealthy.

The Healthy sub-class includes individuals who do not have any recorded medical conditions, ensuring that their profiles remain streamlined within the system.

The Unhealthy sub-class contains information about citizens with medical conditions. To prevent data duplication, this subclass retrieves essential patient details from the Civil Information class. Additionally, all specific disease-related information is managed separately within the Disease class, ensuring a structured and efficient approach to storing and accessing health records.

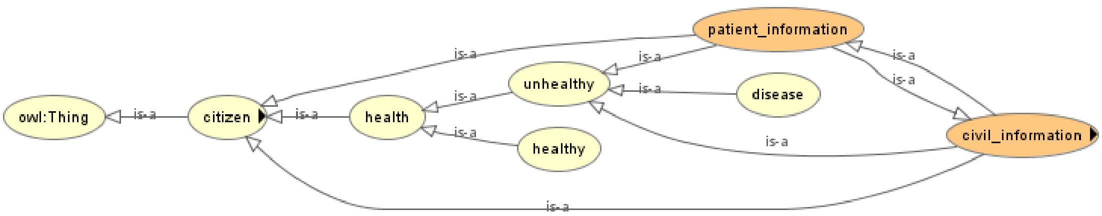

FIGURE 4.6: The health class ontology

The Document class (Figure 4.7) encompasses all essential citizen-related documents, ensuring streamlined access to official records for various administrative processes.

This class includes documents such as the Certificate of Non-Work, School Certificate, Medical File, Vehicle Registration Document, Driving License, Professional Card, Voter Card, Credit Card, Student Card, Identity Card, Commercial Register, Property Contract, Passport, and Healing Card. By centralizing these records within the ontology, government institutions can efficiently manage and verify citizen documents, facilitating seamless service delivery and reducing bureaucratic delays.

FIGURE 4.7: The document class ontology

The Properties class (Figure 4.8) represents all assets owned by a citizen, focusing primarily on real estate and vehicles.

This class provides a structured framework for managing property ownership records, enabling government agencies to track and verify citizen assets efficiently. By integrating property-related data within the ontology, this classification facilitates administrative processes such as taxation, legal documentation, and property registration, ensuring accuracy and transparency in asset management.

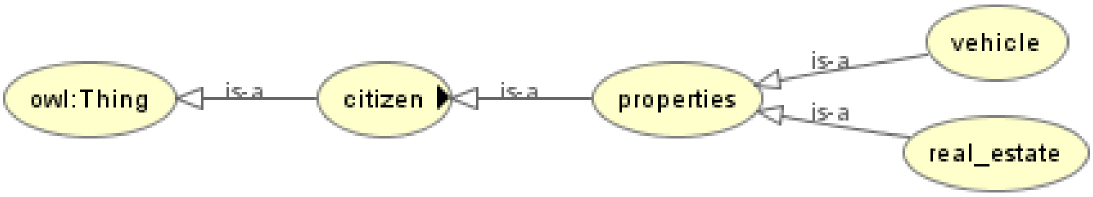

FIGURE 4.8: The properties class ontology

4.3.2.2 Object Property

The Object Property establishes relationships between different entities within the ontology, linking individuals to relevant attributes and classifications. It defines the connections between:

- Citizen and Civil Information – Associating individuals with their personal details.
- Citizen and Education Level – Linking citizens to their academic background.
- Citizen and Job – Defining employment status and professional affiliations.
- Citizen and Health – Connecting individuals to their health records.
- Citizen and Document – Associating citizens with their official documents.
- Citizen and Properties – Identifying assets owned by the citizen.
- Civil Information and Document – Linking personal records with corresponding legal documents.
- Education Level and Document – Connecting academic credentials with official certificates.
- Job and Document – Relating employment records to professional documents.
- Health and Document – Associating medical history with health-related documentation.

- Properties and Documents – Connecting property ownership with relevant legal papers.

These object properties, as outlined in Table 4.2, ensure semantic coherence and interoperability within the ontology, facilitating efficient data retrieval and integration across various governmental services.

TABLE 4.2: Object Properties Table

Domain	**Object Properties**
Citizens	HasInformation, hasLevelEducation, WorkAs, hasHealthInformation, buyMedecinesBy, hasProperty, drivesWith, hasCreditIn, isDefinedBy, OwnsDocument, TravelsWith, VotesBy
Civil information	Isneededin, hasFamillyRecordBook
Education level	Iscausetofind, HasCertificateCard
Job	Isrelatedwith, hasRegistreOf, IsCauseToHave
Health	Hasthesameinformation, HasMedicalFile
Document	Confirms, provesFor, isCardFor, MeansThat, provesForWork, ContainHealthStatus, hasLegalPaperFor
Properties	dependsOn, isRegistredBy, MustHav

4.3.2.3 Data Property

The Data Property defines relationships between entity instances and literal datatype values, ensuring precise data representation within the ontology.

Unlike object properties, which establish links between different entities, data properties connect ontology elements to specific values such as text, numbers, or dates. These properties enable accurate storage, retrieval, and processing of essential citizen information.

Table 4.3 provides a detailed overview of the data properties integrated into our ontology, mapping each entity to its corresponding literal attributes. This structured approach enhances data consistency, validation, and usability across various e-government applications.

TABLE 4.3: Data Properties Table

Class	**Data Properties**
Civil information	HasAdress, hasBirthDate, hasBirthPlace, hasBloodType, HasDivorced, HasFamillyName, HasFatherName, HasFirstName, hasGender, HasMarried, HasMotherFamillyName, HasMotherName, hasNationality, isSingle
Education level	HasHighSchoolField, HasHighSchoolFieldName, HasMiddelSchoolName, hasPrimarySchoolName, hasProfessionalFormation, hasProfessionalFormationField, hasUniversityField, hasUniversityName
Job	hasGovJob, hasPrivateJob, hasProfessionalCard
Health	hasDisease, hasDiseaseDate, hasDiseaseMedicament
Document	HasComercialRegistre, hasCertficateOfNon-Work, HasCreditCard, HasDrivingLicence, hasHealingCard, HasIdCommercialRegistre, HasIdCreditCard, HasIdDrivngLicence, HasIdentityCard, HasIDIdentityCard, hasIdHealingCard, hasIdPassport, HasIDStudentCard, hasPassport, hasPropertyContract, hasStudentCard, hasVehicleRegistrationCard, hasVoterCard
Properties	HasRealestate, hasVehicle

We utilized the Protégé editor to both construct and evaluate our ontology, ensuring its consistency and effectiveness (See Figure 4.9).

4.3.2.4 Validation and Evaluation of The Ontology

The ontology evaluation process was conducted in two key phases:

- Semantic Analysis: This step focuses on identifying the components and semantic relationships between concepts. The evaluation assesses the ontology's ability to accurately define and represent meanings through formulas and relations. To ensure logical coherence, we employed Hermit reasoner, an ontology reasoning tool for validating class hierarchy, detecting subsumption relationships, and verifying the consistency of the OWL-based ontology.

- Semantic Application: In this phase, the ontology was tested against new place names and semantic terms that were not used during its initial design. Additionally, it was assessed using a new semantic file to evaluate its lexical and data layers, focusing on concept representation, vocabulary consistency, and contextual application.

Through this two-step evaluation process, we ensured that the ontology maintains semantic accuracy, logical consistency, and adaptability for future expansions and real-world applications.

In practice, our assessment was conducted using the Hermit reasoner, which enabled us to verify the consistency of the ontology, identify subsumption relationships between classes, and perform additional logical checks.

The evaluation process did not impose any syntactic or semantic constraints, allowing for a comprehensive validation of the ontology's structure and relationships. As a result, we can confidently assert that our proposed ontology is logically sound, free from anomalies, and fully functional.

This validated ontology serves as a robust foundation for interaction, integration, and composition within government electronic services. It will play a crucial role in supporting the future development of our e-government framework, ensuring seamless interoperability across various administrative systems.

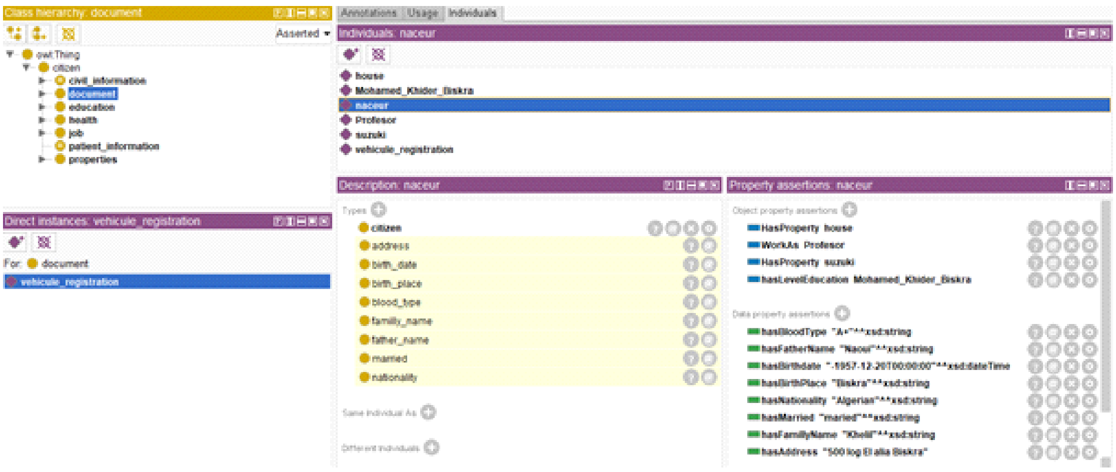

FIGURE 4.9: Protégé view of the citizen's ontology

4.3.3 Part 2 Conclusion

In this part, we explored the challenges of e-government adoption in Algeria, highlighting the fragmented development of digital applications across various government departments. A key issue identified is the lack of interoperability between services, even within the same organization, which hinders efficient public administration.

To address this, we developed a specialized ontology designed to support decentralized administrative procedures, a crucial aspect of Algeria's electronic governance strategy. Our ontology encompasses civil, health, education, and professional information, ensuring a structured and standardized representation of citizen data.

The research is rooted in Semantic Web technologies, utilizing OWL (Web Ontology Language) to enhance semantic expressiveness and logical relations. For implementation, we selected Protégé 5.5.0, a widely used ontology development tool supported by METHONTOLOGY, known for its platform independence and extensibility. Additionally, we employed Hermit reasoner to validate the ontology, ensuring its consistency and correctness.

Our ontology can be further expanded to align more closely with the needs of government and administrative institutions by incorporating new classes, refining object properties, and adjusting data relationships. Moving forward, this ontology will serve as a foundational component of our broader framework, which aims to facilitate electronic service integration and interaction through Web service composition, ultimately improving Algeria's e-government ecosystem.

4.4 Part 3: QoS-aware Service Composition Problem Based on an Improved BTLBO Algorithm

To improve interactions between the government and citizens, public organizations are increasingly offering online services. However, a single service is often insufficient to fully meet citizens' diverse needs. The integration of multiple web services, known as "web service composition," allows for more comprehensive solutions. Web service composition is a key aspect of modern e-government systems. It involves creating new value-added services by combining existing ones. This not only reduces the development time and costs of new applications but also significantly improves interoperability and fosters greater collaboration between administrative entities.

However, with the convergence of a large number of web services with the same function, quality of service (QoS) has increasingly become an essential factor that needs to be considered in the process of selecting services from these function-equivalent services for composition. Therefore, two approaches must be distinguished, in the QoS-aware service composition. In the first one, a predefined workflow is supposed to be known. This workflow describes a set of abstract task to be performed. Moreover, associated to each task, a set of Wss with the similar functionalities (but different QoS) is also known. The composition problem is then to select one WS per task to respect QoS objective. In the second approach, the existence of a predefined workflow is not assumed. Discovery and connection between WS are automatically performed by syntactic and/or semantic matching.

In general, finding a composite web service that meets a client's QoS requirements is an NP-hard problem. According to a recent survey [132], its solution takes exponential time and cost. Meta-heuristic approaches provide possible solutions to the QoS-aware service composition problem [132, 134].

Meta-heuristics are computational intelligence paradigms for solving NP-hard optimization problems, they are based on an iterative process that is guided by heuristics using concepts to intelligently browse the search space, and learn strategies to structure the information in order to efficiently find near-optimal solutions, in a reasonable time [135].

Among the meta-heuristics that dominate the field of WSs [132, 134] are the Genetic Algorithm (GA) and the Particle Swarm Optimization particle (PSO). The works in this context, are distinguished by the definition of the objective function, the number and type of QoS, the coding of the solution and the operators used in the evolution of solutions.

Despite the abundance of works based on Meta-heuristics, in the literature [132, 134], the problem of high computational complexity of service selection remains posed by the research community. They state that the performance of Meta-heuristic algorithms, in terms of parameters that affect the efficiency of the algorithms, has not yet been properly explored.

In the following, we introduce an improved approach for identifying composite web services that meets a client's Quality of Service (QoS) requirements [136]. To effectively tackle the discrete and combinatorial nature of the service selection problem, we propose a discrete adaptation of the Balanced Teaching-Learning-Based Optimization (BTLBO) algorithm [137], referred to as DBTLBO (Discrete BTLBO). This tailored version is specifically designed to optimize the selection of web services within an e-government context, ensuring that the resulting compositions meet both functional and non-functional requirements.

4.4.1 QoS-aware Service Composition Problem

We consider a user request consisting of n service tasks (abstract services), denoted as $S_t = A_{S_1}, ..., A_{S_i}, ..., A_{S_n}$. Each abstract service $A_{S_i} \in S_t$, has m candidate atomic services (concrete services) $Cs_{i,j}$, j $\in [1, m]$, with the same functions and different QoS values Q_k, k $\in [1, l]$, l is the number of QoS value criteria.

In our work, we focus on five quantitative non-functional properties of web services which are: Q_1: Response Time, Q_2: Reputation, Q_3: Price, Q_4: Reliability, and Q_5: Availability. As long as there are several potential web services for a task or process, alternative combinations of web services with the same functionality but distinct qualities will exist, as illustrated in Fig4.10.

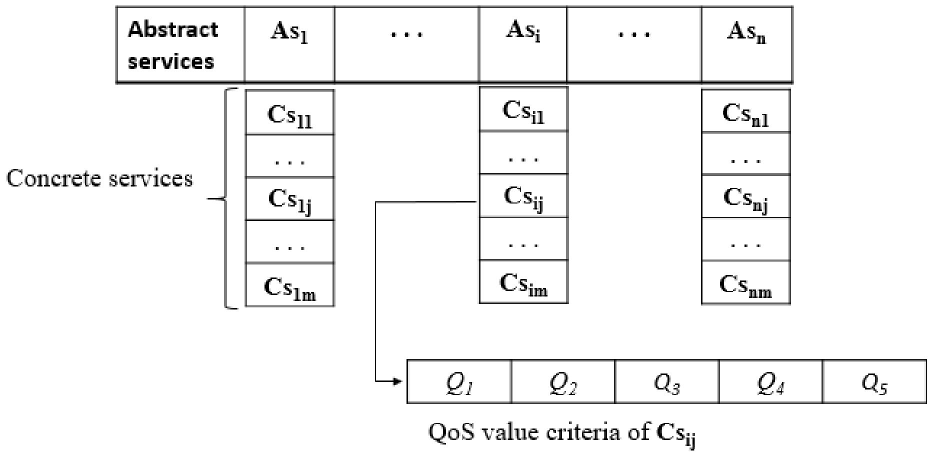

FIGURE 4.10: QoS-aware Service Composition Problem

The goal of QoS-aware service composition is to determine an optimal composition plan that meets a client's QoS requirements, making it a combinatorial optimization problem.

This process involves instantiating abstract tasks with concrete services while optimizing a set of QoS criteria and ensuring that all global constraints (Qc_1, Qc_2, Qc_3, Qc_4, Qc_5) are satisfied. Each Qc_k (k = 1...5) represents the client's QoS requirement for the corresponding QoS criterion Q_k across n service tasks. Given the exponential number of possible compositions (m^n for n tasks and m candidate services per task), this problem is classified as NP-Hard.

In order to evaluate the QoS values of n concrete services representing composite services X = (x_1, . . ., x_n), we applied two types of QoS aggregation functions [192]: the summation for the price, the reputation and the response time. The multiplication for the reliability and availability (see Table 4.4).

TABLE 4.4: QoS Aggregation Functions

QoS Criteria	Aggregation Function
Response Time	$q_1 = \sum_{i=1}^{n} Q_1(x_j)$
Price	$q_2 = \sum_{i=1}^{n} Q_2(x_j)$
Reputation	$q_3 = \frac{1}{n}\sum_{i=1}^{n} Q_3(x_j)$
Reliability	$q_4 = \prod_{i=1}^{n} Q_4(x_j)$
Availability	$q_5 = \prod_{i=1}^{n} Q_5(x_j)$

To formalize the QoS-aware service composition problem, we need to find a fitness function that maximizes the QoS value function F and minimizes the number of constraint violations of composite services.

To define the quality of service (QoS) value function F, we employ a weighted sum approach that aggregates various QoS criteria. This method considers both negative criteria (e.g., price, response time) and positive ones (e.g., reputation, reliability, availability). While positive criteria are intended to be maximized, negative criteria must be minimized. To unify the evaluation, all negative QoS values are multiplied by -1, thereby converting the optimization objective into a maximization problem for all criteria. Under this formulation, the QoS value function $F(X)$ for a composite service $X = (x_1, \ldots, x_n)$ is defined as follows:

$$F(X) = \sum_{k=1}^{2} w_k \left(\frac{Q_k^{max} - q_k(X)}{Q_k^{max} - Q_k^{min}} \right) + \sum_{k=3}^{5} w_k \left(\frac{q_k(X) - Q_k^{min}}{Q_k^{max} - Q_k^{min}} \right) \tag{4.1}$$

Where q_k is the aggregate QoS value of criterion k; Q_k^{max}, Q_k^{min} represent the possible maximal and minimal aggregated QoS values of criterion k, with $Q_k^{max} \neq Q_k^{min}$, w_k is the weight for QoS criterion k and $\sum_{k=1}^{5} w_k = 1; 0 < w_k < 1$

To incorporate global constraints in the evaluation of a composite service $X = (x_1, \ldots, x_n)$, several studies [10, 193, 194] have proposed the use of penalty functions within the fitness function formulation. The purpose of these penalty functions is to reduce the fitness value whenever one or more global constraints are violated.

In our work, we adopt the total number of constraint violations in X, as a penalty function $P(X)$. In this case, the used fitness function *fit* combines both the QoS value function $F(X)$ and the penalty function $P(X)$ in the same expression as follow:

$$fit(X) = F(x) + P(x) \tag{4.2}$$

where

$$P(X) = -\sum_{k=1}^{R} (D_k)^2(X) \tag{4.3}$$

and

$$D_k(X) = \begin{cases} 0 & \text{If } q_k(X) \geq Qc_k. \\ |q_k(X) - Qc_k| & \text{otherwise} \end{cases} \tag{4.4}$$

Where $F(X)$ is the QoS value function, $P(X)$ is the penalty function, $q_k(X)$ is the aggregate QoS value of criterion and Qc_k is the client's QoS requirement imposed on QoS value of Q_k criterion of the composite service X.

4.4.2 Meta-heuristic Approaches for QoS-aware Service Composition Problem

Research on Web service composition has garnered significant attention, with existing approaches generally categorized into automated Web service composition and QoS-aware service composition.

QoS-aware service composition is particularly well-suited for applications with dynamic requirements, where service availability and capabilities change over time. The objective of these approaches is to select the most suitable candidate services based on user-specific requirements, ensuring that the final composite service meets global QoS constraints.

Meta-heuristic algorithms have emerged as powerful solutions for tackling web service composition problems, enabling efficient service selection under complex constraints. These approaches are categorized into two main types:

- Mono-Objective Meta-heuristics – Focus on optimizing a single objective function. They are further classified into:

  - Local Selection Models: Select one service per class using an objective function, independently of other service classes, before composing the final solution [150, 151, 154].
  - Global Selection Models: Utilize a single objective function, often represented as a weighted sum of QoS attributes, to optimize the overall service composition [138–140, 153, 155].
  - Hybrid Models: Combine elements of both local and global selection models, offering reduced complexity while effectively handling global constraints [141–143].

- Multi-Objective Meta-heuristics – These approaches optimize multiple QoS attributes simultaneously, typically using the Pareto optimality concept. This reduces the number of candidate service implementations, with Pareto front solutions representing the most optimal service selections [129, 144–146].

The web service selection problem in Meta-heuristic approaches can be represented using two primary models:

- Graph Model: Represents web services as nodes, with edges connecting two services. The edge weight denotes the QoS characteristics of the link between two services [154].

- Combinatorial Model: Encodes a solution as an integer array, where each element corresponds to a selected concrete service for an abstract task [153, 156]

Among the most widely used Meta-heuristics in QoS-aware service composition are the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) [132, 134]. Research in this field varies based on objective function formulation, QoS parameters, solution encoding techniques, and evolutionary operators used in the optimization process.

The efficiency of Meta-heuristic algorithms is largely determined by their convergence rate and execution time, making it crucial to balance exploration and exploitation in order to achieve optimal service selection.

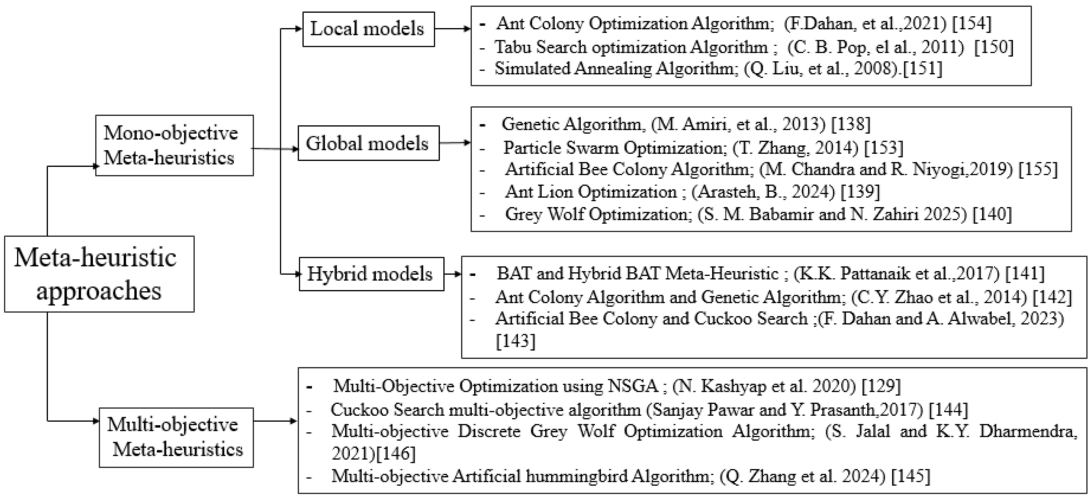

FIGURE 4.11: Meta-heuristic models for QoS-aware service composition problem

4.4.3 Discrete Balanced Teaching-learning-based Optimization Algorithm for QoS-aware Service Composition Problem

To address the challenge of selecting an optimal web service composition based on non-functional quality of service (QoS) criteria from a set of candidate services, while meeting specific user requirements and preferences, we propose a discrete balanced teaching-learning optimization (DBTLBO) algorithm.

The proposed algorithm is a discrete adaptation of the original balanced teaching-learning optimization (BTLBO) algorithm [137], which is itself an improvement on the standard teaching-learning optimization (TLBO) algorithm [175], designed to achieve an efficient balance between exploration and exploitation through self-adaptive tuning of control parameters. The algorithm operates in four main phases: teaching, learning, tutoring, and restarting, each contributing distinctly to the optimization process.

The teaching phase models the transfer of knowledge from the teacher to the students (i.e., from the best solution to the other candidate solutions in the population). To improve exploitation capacity while preserving solution diversity, this phase uses a weighted average, unlike the standard average used in TLBO, to guide learning. The weighted average reflects the difference between a learner's performance and the average performance of all learners, allowing for more targeted improvements.

The learning phase simulates learning through interaction with other learners. This phase is identical to that of the original TLBO algorithm, where individuals engage in mutual learning, allowing a learner to acquire knowledge from another randomly selected peer with a higher score.

The tutoring phase simulates private teaching, which is absent from the TBLO algorithm. It involves improving the quality of the solution and accelerating convergence through intensive local search around the best historical solution, thus strengthening the algorithm's ability to exploit promising areas of the search space.

Additionally, the Restarting phase, also absent in the original TLBO, simulates the re-enrollment process in subjects where learners performed poorly. This mechanism is triggered after all other learning opportunities have been exhausted and prevents premature convergence by helping the algorithm escape local optima and continue its progress toward the global optimum.

4.4.3.1 Operating principle of the DBTLBO Algorithm

The DBTLBO algorithm is a population-based Meta-heuristic that simulates four learning strategies within a classroom setting. In the context of web service selection, the classroom of students (learners) is interpreted as a population of candidate solutions. Each learner corresponds to a composite service representing a potential solution. The various subjects studied by the learners are analogous to decision variables, which reflect the selection of concrete services for the given abstract services. The performance of a learner is evaluated using a fitness value, comparable to the quality score of a composite service (see Equation 4.2). The best-performing solution within the population is designated as the teacher—typically considered the most knowledgeable individual responsible for improving the performance of others.

The DBTLBO algorithm proceeds as follows: After initializing the population and associated parameters, the search process begins by identifying the teacher (i.e., the best solution in the current population). Subsequently, in each iteration, one of three learning phases—*Teaching*, *Learning*, or *Tutoring*—is randomly selected for all learners. This is followed by a *Restarting* phase. The detailed implementation of DBTLBO for the web service selection problem is described below:

Initialization of the Optimization Parameters

The DBTLBO algorithm requires the initialization of only two control parameters: the number of learners in the classroom, denoted as *Nl*, which corresponds to the population size, and the maximum number of function evaluations, denoted as *MaxNE*, which serves as the termination criterion. All other parameters are adaptively adjusted during the execution of the algorithm. (See table 4.5)

TABLE 4.5: Parameters and Definitions

Parameters	**Definitions**
n	Number of abstract tasks required by the user (number of decision variables).
M	Number of concrete services for each abstract task.
Bs	A service base segmented into n classes (each service is characterized by 5 QoS values).
Nl	Number of solutions (learners).
Ce	Counter of function evaluations.
$MaxNE$	The maximum number of function evaluations.
$MaxF$	Maximum number of allowable failings.
Cfi	Counter of failings for learner i.
Tnf	The total number of failing learners in the population.
Nf	Number of failings.

Individual Representation and Initialization

Initialize the classroom of learners: Generate a random population P of *Nl* learners, P = ($X_1, X_2, ..., X_i, ..., X_{Nl}$), each learner is a solution $X_i = (x_{i,1}, x_{i,2}, ..., x_{i,j}, ..., x_{i,n})$ that denotes a composed service. It is formed of n decision variables that denote the concrete service codes. $X_{i,j}$ is the code of concrete service that is selected for the j^{th} abstract service.

$$P(X)=\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,j} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,j} & \dots & x_{2,n} \\ & & \dots & & & \\ x_{i,1} & x_{i,2} & \dots & x_{i,j} & \dots & x_{i,n} \\ x_{Nl,1} & x_{Nl,2} & \dots & x_{Nl,j} & \dots & x_{Nl,n} \end{bmatrix}$$

Concrete services are encoded using an integer-based representation. For each abstract service, the available concrete services are indexed starting from 1. Thus, if an abstract service has M concrete services, they are assigned codes from 1 to M, i.e., [1, 2, 3, ..., M]. For instance, the solution $X_i = (2, 1, 5, 2, 4, 3)$ (as illustrated in Figure 4.12) indicates the selection of concrete service 2 from the first abstract service, concrete service 1 from the second, concrete service 5 from the third, concrete service 2 from the fourth, concrete service 4 from the fifth, and concrete service 3 from the sixth abstract service.

2	1	5	2	4	3

FIGURE 4.12: Solution coding example

Evaluation of the Solutions

Each solution X_i in the population is evaluated using the fitness function *fit*, as defined in Equation (4.2). Based on this evaluation, the solution with the highest fitness value—denoted as *Xbest*—is selected to serve as the teacher. This teacher solution is represented as $X_t = (xt_1, xt_2, \ldots, xt_j, \ldots, xt_n)$. Subsequently, the mean knowledge of all learners (i.e., the mean solution) is computed and denoted by $M = (m_1, m_2, \ldots, m_j, \ldots, m_n)$, where each m_j represents the average value for the j^{th} decision variable across the population.

$$m_j = \frac{1}{Nl} \sum_{i=1}^{Nl} X_{i,j}, j = 1...n \tag{4.5}$$

Teaching Phase

In this phase, each learner $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,n})$ in the population updates its position based on the teacher's position $X_t = (xt_1, xt_2, \ldots, xt_j, \ldots, xt_n)$ and a weighted mean vector $W_m = (w_{m1}, w_{m2}, \ldots, w_{mj}, \ldots, w_{mn})$, which represents an intermediate position between X_i and the population mean $M = (m_1, m_2, \ldots, m_j, \ldots, m_n)$.

To promote diversity in the search process, a different learner $X_k = (x_{k,1}, x_{k,2}, \ldots, x_{k,j}, \ldots, x_{k,n})$, where $k \neq i$, is randomly selected from the population. The weighted mean vector W_m is then computed as follows:

$$wm_j = \frac{m_j + (\Theta \times x_{i,j}) + (1 - \Theta)x_{k,j}}{2}; j = 1...n \tag{4.6}$$

where Θ is a random value that can be 0 or 1, as indicated by:

$$\Theta\text{=round(rand(1,0))}$$

The updated position of learner $X_i = (x_{i,1}, x_{i,2}, ..., x_{i,j}, ..., x_{i,n})$ is calculated as:

$$nx_{i,j} = x_{i,j} + \Theta \times r \times T_f \times (xt_j - wm_j) + (1 - \Theta) \times r \times (xt_i - T_F \times wm_j); j = 1...n \tag{4.7}$$

where, $nx_{i,j}$ is the j^{th} decision variable of learner update position X_i, and $x_{i,j}$ is the j^{th} decision variable of learner current position X_i. xt_j is the j^{th} decision variable of the teacher Xt's position.r is a single random value between [0–1], T_F is the teaching factor, its value can be either 1 or 2, and it is decided randomly with equal probability as:

$$T_F\text{=round[1+rand(0,1)\{2-1\}]}$$

The parameter T_F plays a key role in balancing intensification and diversification within the search process. When $T_F = 1$, the Teaching phase emphasizes exploration by targeting the best region found so far (diversification). In contrast, when $T_F = 2$, the phase focuses on exploiting the current best solution (i.e., the Teacher), thereby enhancing intensification.

Given that the Web service composition problem is a discrete and combinatorial optimization task, each decision variable $nx_{i,j}$ must take an integer value from the set $\{1, 2, \ldots, m\}$. Consequently, a rounding operator is applied after computing each $nx_{i,j}$ to ensure that the result remains within the valid discrete domain, as shown in the following equation:

$$nx_{i,j} = round(nx_{i,j}) = \begin{cases} 1 & \text{if } nx_{i,j} < 1 \\ k, k \in \{1, 2, ..., m\} & \text{if } 1 \le nx_{i,j} \le m, j = 1, ..., n \\ m & \text{if } nx_{i,j} > m \end{cases} \tag{4.8}$$

Learning Phase

The objective of the Learning phase is to enhance the knowledge of a learner X_i by enabling interaction with another randomly selected learner X_k. Through this interaction, the learner X_i updates its position based on the comparative performance of the two solutions. The new position nX_i of learner X_i is generated as follows:

$$nx_{i,j} = \begin{cases} x_{i,j} + r(x_{i,j} - x_{k,j}) & \text{if } fit(X_i) > fit(X_k) \\ x_{i,j} + r(x_{k,j} - x_{i,j}) & \text{otherwise } j = 1, 2, ..., n \end{cases} \tag{4.9}$$

Here, $nx_{i,j}$ represents the j^{th} decision variable of the updated position for learner X_i, while $x_{i,j}$ denotes the j^{th} decision variable of the current position of the same learner. Similarly, $x_{k,j}$ corresponds to the j^{th} decision variable of the position of learner X_k, which is randomly selected from the population. The variable r is a random vector with values uniformly distributed in the interval $[0, 1]$. The functions $fit(X_i)$ and $fit(X_k)$ represent the fitness values of learners X_i and X_k, respectively.

Tutoring Phase

The Tutoring phase acts as a local search operator aimed at enhancing the quality of learners by refining their solutions. It performs effective exploitation by replacing a subset of a learner's decision variables with the corresponding variables from the best solution found so far. During this phase, a subset Sp consisting of k decision variables is randomly selected. The size k of the subset Sp is determined as follows:

$$k = n \times r \times (e^{-(1-\alpha)^2}) \tag{4.10}$$

Where r: is a single random value between [0–1];

n: is the number of decision variables;

α: indicates the evaluation rate, which affects the number of decision variables to update during the local search of the Tutoring phase. It represents the ratio between the function evaluation counter *Ce* and the maximum number of function evaluations *MaxNE*, as shown by the following equation:

$$\alpha = \frac{Ce}{MaxNE} \tag{4.11}$$

Where k is the number of decision variables to be updated. *Ce* is the counter of function evaluation, *MaxNE* is the maximum number of function evaluations,

Therefore, the nX_i^{Sp} as the updated version of learner X_i, is calculated as follow

$$nX_i^{Sp} = Xt^{Sp} + U^{Sp}(-1,+1) \times (1-\alpha)^2 \times (X_i^{Sp} - X_j^{Sp}) \tag{4.12}$$

Where, Xt^{Sp} is the *Sp* of the teacher position, U^{Sp} (-1,+1) is a random vector uniform distribution between [-1 +1], X_i^{Sp} is the selected *Sp* of learner current position X_i, and X_j^{Sp} is the selected *Sp* of learner X_j which is randomly chosen from the population.

The rounding operator (eq4.8), was applied on the modified decision variables to discretize them

Restarting Phase

The restart phase is triggered once all learners have successfully completed one of the randomly selected learning phases (teaching, learning, or tutoring). To enhance exploration, inactive learners who do not surpass a predefined fitness threshold, denoted as "*Fth*", should undergo a partial or complete restart during this phase.

The value of failing threshold *Fth* directly relates to maximum number of allowable failings (*MaxnF*) and calculated as:

$$Fth = (MaxF \times \alpha^2) \tag{4.13}$$

$$MaxF = \frac{MaxNE}{20 \times n} \times e^{\left(\frac{1}{MaxNE}\right)^{\frac{n}{100}^2}} \tag{4.14}$$

where, *MaxF* is the maximum number of allowable failings, *MaxNE* is the maximum number of function evaluations, n is the number of decision variables, and α is calculated as eq. 4.11

We define Cf_i to represent the number of failures of learner X_i. If Cf_i exceeds the threshold *Fth*, learner X_i must be restarted. This implies that the *k* decision variables of learner X_i are randomly selected and replaced by values randomly chosen within their defined intervals. The value of *k* is determined according to equation 4.10. However, if more than 50% of learners fail for a duration of $MaxnF/10$, the entire population should be restarted, and the parameter α must be reset.

The execution of the restart procedure is associated with a maximum allowable number of failures, denoted as "*MaxF*" (the maximum number of accepted failures), the failure counter for each learner X_i, represented by Cf_i, the counter for function evaluations, denoted as *Ce*, and the total number of failing learners in the population, represented by *Tnf*.

Algorithm 1 DBTLBO

Require: *Bs*: A service base segmented into n classes (each service is characterized by 5 QOS values).*Nl*: number of solutions (learners).*MaxNE* is the maximum number of function evaluations;

Ensure: The best composed services "*BestComp*" and its fitness value "*best-fit*";

`Generate a random population P of Nl solutions`
Choose a random solution from P, and store it as *BestComp* and its fitness value into *best-fit*;
Evaluate each solution $X_i = (x_{i,1}, x_{i,2}, ..., x_{i,j}, ..., x_{i,n})$ *i*=1...*Nl*, as given in Eq.4.2;
Put *Tnf*=0;
Put $Cf_i = 0$; *i*=1...Nl;
Ce=1;
while Ce $\leq$ *MaxNE* **do**
    Calculate the parameter α according to Eq4.11;
    Calculate the failing threshold *Fth* according to Eq4.13;
    Find the best solution "*Xbest*" in the population *P*;
    Put *Xbest* as a teacher $Xt = (xt_1, xt_2, ..., xt_n)$;
    if $F(Xbest) > F(BestComp)$ **then**
        replace *BestComp* with *Xbest* and store its fitness value into *best-fit*
    end if
`For each` learner X_i, *i*=1...*Nl*, in population `do`
    Randomly choose a learner $X_k; i \neq k$;
    Choose a phase randomly from *Ph* $\in$ {1,2,3};
`Switch Ph`
    Case 1: Teaching Phase
    Calculate the new solution $nX_i of X_i$ according to Eq.4.6 and Eq.4.7;
    Case 2:Leaning phase
    Calculate the new solution $nX_i of X_i$ according to Eq.4.9;
    Case 3:Tutoring phase
    Calculate the new solution $nX_i of X_i$ according to equations Eq.4.10, Eq.4.11 and Eq.4.12;
`EndSwitch`
    Apply the rounding operator, according to the Eq.4.8;
    Evaluate the solution nX_i according to Eq.4.2;
    if $fit(nX_i) > fit(X_i)$ **then**
        Replace X_i by nX_i; (Accept nX_i)
        Cf_i=0;
        Replace X_i by nX_i if $fit(nX_i) > fit(X_i)$; (Update the teacher solution)
    else
        Reject nX_i;
        $Cf_i = Cf_i + 1$; (count the failings of learner i)
        Tnf= *Tnf*+1; (count the total number of failing learners)
    end if
    Ce = *Ce* + 1; (to count the function evaluation)
`EndFor`
    Execute the Restarting procedure(Algo.2);
end while
Return *BestComp* and *best-fit*;

Algorithm 2 Restarting procedure

Input: *P*, *MaxF*, Cf_i, *Ce*, *Tnf*;

Output: Restarted population;

1: Update the value of α according to Eq4.11
2: Update *Fth* according to Eq4.13
3: **if** (Ce $\geq \frac{MaxNE}{10}$)and (Tnf $> \frac{Nl}{2}$) **then** (The whole population restarting)
4: Generate a random population *P* of *Nl* solutions;
5: `For` i:=1 `to` *Nl* `do`
6: Evaluate a solution X_i, from *P*, using the fitness function *fit* as given in Equation (4.2);
7: *Ce*=*Ce*+1;
8: $Cf_i = 0$;
9: `EndFor`
10: *Tnf*=0;
11: **else** (Individual Restarting)
12: `For` i:=1 `to` *Nl* `do` (For all Learners)
13: **if** $Cf_i > Fth$ **then** (Learner X_i fails more than a predefined failing threshold)
14: Calculate the number of decision variables *K* to be restarted as eq4.10 and eq4.11;
15: Choose *K* random decision variables from the solution X_i and replace them with random values chosen from the interval [1 *n*];
16: Evaluate the solution X_i according to Eq4.2;
17: *Ce*=*Ce*+1;
18: $Cf_i = 0$;
19: **end if**
20: `EndFor`
21: **end if**
22: Return the restarted population;

4.4.3.2 Time Complexity of the DBTLBO

The time complexity of the DBTLBO algorithm is influenced by the following parameters: the maximum number of function evaluations, denoted as *MaxNE*, the number of learners *Nl* (i.e., the number of solutions), and the number of abstract services, *n*, which corresponds to the size of the solutions (learners). As described in algorithm 2, the time complexity for evaluating solutions (as seen in line 3) is *n* * *Nl*. In each iteration (as shown in lines 7 - 40), five components contribute to the overall complexity:

- The time complexity for calculating the new solution (during the teaching, learning, or tutoring phase) is $2n$ computation times (lines 15 - 25).
- The time complexity for calculating the rounding operator is n computation times (line 26).
- The time complexity for evaluating solution fitness is n computation times (line 28).
- The time complexity of the Restarting phase is n computation times.

Therefore, the overall time complexity of the DTLBO-WS algorithm is:

$$n * Nl + MaxNE(Nl * n + 5n) \text{ computation times} = O(MaxNE * Nl * n),$$

where $MaxNE$ is the maximum number of function evaluations, Nl is the number of learners (solutions), and n is the number of abstract services.

4.4.3.3 Experiment and Analysis

To evaluate the performance of DBTLBO, we conducted simulations using MATLAB 2018a on a 64-bit Windows operating system with an Intel Core i5-5200U processor (2.2 GHz) and 12 GB of RAM. A web service dataset was created to test the effectiveness of our method across different problem space sizes. The dataset consists of 500 abstract service classes, with 500 concrete services associated with each abstract service class, as shown in Table 4.6.

Table 4.6: Simulation Settings of Abstract and Concrete Service Numbers

Simulation Number	Abstract Services	Concrete Services
1	100	Varying from 50 to 500
2	Varying from 50 to 500	100

As suggested in [195], all QoS criterion values for concrete services are uniformly distributed within a specified interval (see Table 4.7). The user preference weights for response time, price, reputation, reliability, and availability, used in the experiments, are 0.2, 0.3, 0.1, 0.2, and 0.2, respectively.

To demonstrate the capabilities of the DBTLBO algorithm in improving QoS-aware service composition, it is compared to the TLBO, GA, and PSO algorithms using the

TABLE 4.7: QoS Criteria Considered for Simulation

Type	**QoS Criteria**	**Interval**
Negative	Response Time	[0 – 300]
	Price	[0 – 30]
Positive	Reputation	[0 – 5]
	Reliability	[0.7 – 1]
	Availability	[0.5 – 1]

same dataset and user preference weights for QoS criteria, as shown in Table 4.7. To ensure a fair comparison, all algorithms are implemented in the same environment with the same maximum number of function evaluations and population size. The results are presented in terms of the mean and standard deviation (SD). We calculated the average fitness value and the standard deviation after repeating each program execution 20 times.

The Genetic Algorithm (GA) is an iterative optimization technique that starts with a population of randomly initialized solutions, where each solution represents a list of potential web services (WSs) for the composite service. The fitness value of each solution is calculated using Equation (Eq.4.2). In each iteration, the roulette wheel selection algorithm is applied to select parent solutions. A single-point crossover operation is performed on the parent solutions with an 80% rate to generate the children population, followed by a mutation operator with a 5% rate applied to the children to generate the new population. The best solution, having the maximum fitness value among all the generated solutions, is selected. The evolutionary parameters used are inspired by [196], where experiments on crossover and mutation configuration parameters yielded the best score for the utility function.

The Particle Swarm Optimization (PSO) algorithm [197] is a population-based search algorithm that solves optimization problems by generating a population of candidate solutions, called particles, which move through the search space according to their positions and velocities. The position of each particle determines its fitness value, while its velocity controls the direction and distance of its search. In PSO, the particle population is initialized randomly, with each particle's position representing a list of candidate WSs for the composite service. The fitness value for each particle is calculated using Equation (Eq.4.2). The best particle with the maximum fitness value is designated as "*gbest*", while the best position found by the particle itself during its previous movements is denoted "*pbest*". PSO involves several parameters—namely inertia weight, cognitive learning factor, and social learning factor—which are used to update the velocity and position of particles. For implementing PSO, we adopted values inspired by [198], with inertia weight set to 0.7, cognitive learning factor $C_1 = 2$, and social learning factor $C_2 = 2$. The rounding operator, as shown in Equation (Eq.4.8), is used to discretize the updated positions.

Two experiments were conducted to evaluate the overall fitness values of the competing algorithms. In the first experiment, the number of concrete web services for each abstract web service ranged from 50 to 500, with increments of 50. The number of abstract web services was fixed at 100. In the second experiment, the number of abstract web services ranged from 50 to 500, with increments of 50, while the number of concrete web services for each abstract web service was fixed at 200. The simulation results are presented in Fig. 4.13, Fig. 4.14, Table 4.8, and Table 4.9. The best overall results are highlighted in bold.

Table 4.8: Fitness Evaluation According to Number of Concrete Services (CS)

CS	**DBTLBO**		**TLBO**		**PSO**		**GA**	
	Avg-fit	Div (%)	Avg-fit	Div (%)	Avg-fit	Div (%)	Avg-fit	Div (%)
50	0.91	0.02	0.87	0.14	0.80	0.16	0.76	0.33
100	0.92	0.01	0.86	0.21	0.81	0.11	0.77	0.24
150	0.91	0.02	0.88	0.13	0.88	0.24	0.84	0.16
200	0.90	0.10	0.87	0.09	0.84	0.33	0.78	0.27
250	0.92	0.03	0.90	0.11	0.86	0.15	0.76	0.45
300	**0.95**	**0.01**	0.86	0.33	0.82	0.27	0.80	0.25
350	0.93	0.15	0.87	0.24	0.87	0.15	0.77	0.67
400	0.94	0.04	0.89	0.17	0.84	0.23	0.82	0.31
450	0.93	0.11	0.88	0.26	0.82	0.31	0.83	0.16
500	0.91	0.01	0.87	0.18	0.86	0.14	0.84	0.29

Table 4.9: Fitness Evaluation According to Number of Abstract Services (AS)

AS	**DBTLBO**		**TLBO**		**PSO**		**GA**	
	Avg-fit	Div (%)	Avg-fit	Div (%)	Avg-fit	Div (%)	Avg-fit	Div (%)
50	0.93	0.05	0.88	0.17	0.85	0.18	0.80	0.15
100	0.94	0.01	0.87	0.16	0.81	0.13	0.77	0.21
150	0.89	0.01	0.86	0.04	0.83	0.10	0.81	0.33
200	0.91	0.03	0.87	0.10	0.84	0.16	0.78	0.18
250	0.93	0.00	0.85	0.16	0.86	0.21	0.79	0.09
300	0.90	0.02	0.88	0.21	0.84	0.06	0.83	0.15
350	**0.95**	**0.02**	0.90	0.16	0.87	0.15	0.77	0.11
400	0.93	0.01	0.86	0.10	0.85	0.14	0.80	0.17
450	0.92	0.10	0.84	0.12	0.81	0.34	0.82	0.19
500	0.94	0.06	0.85	0.13	0.84	0.27	0.82	0.24

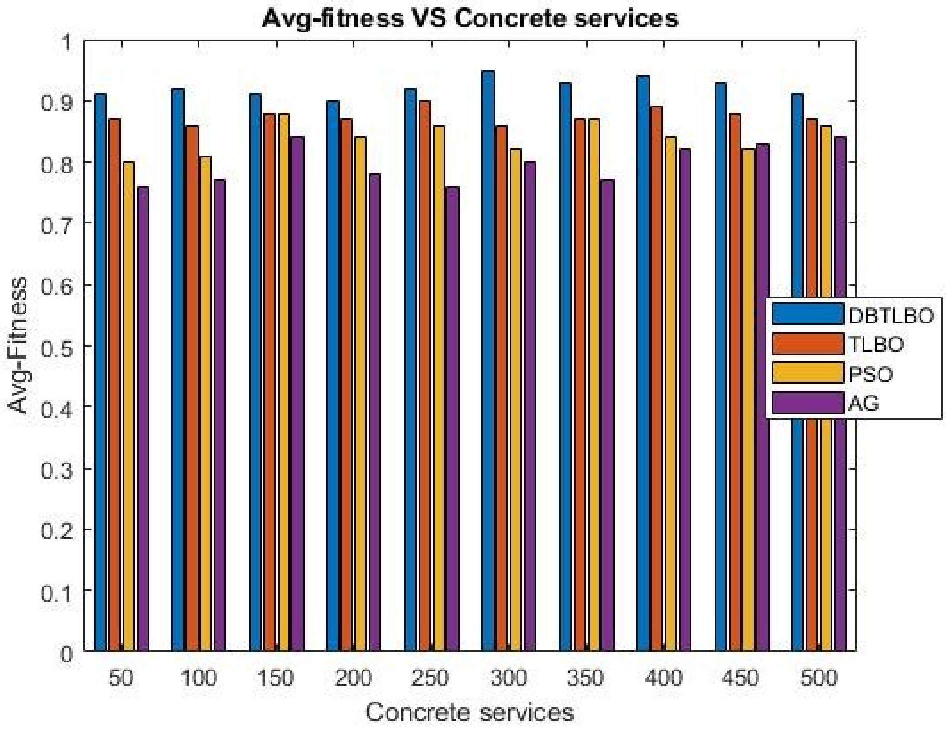

FIGURE 4.13: Fitness evaluation according to number of abstract services

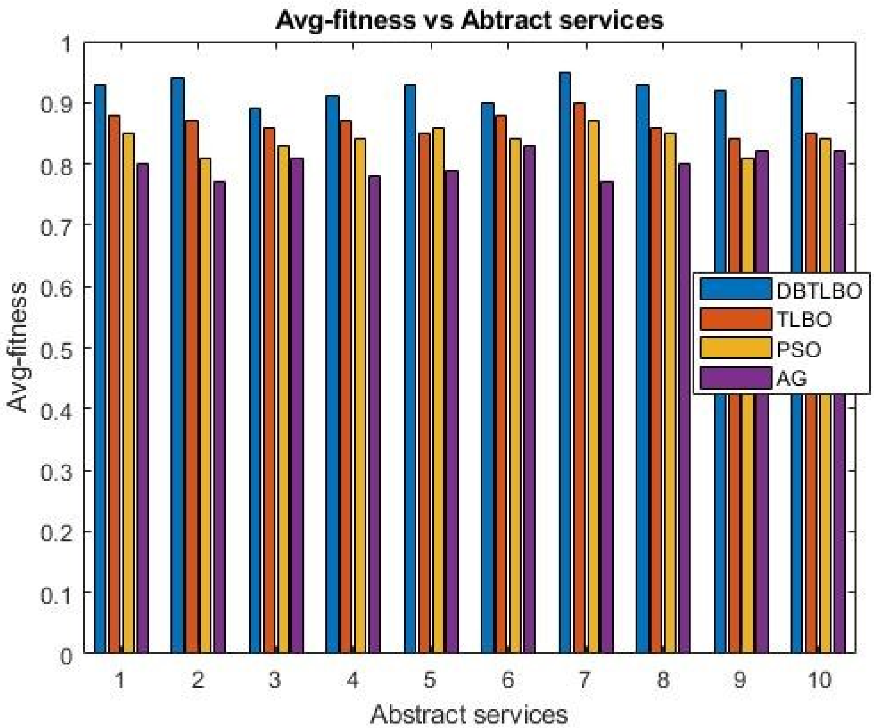

FIGURE 4.14: Fitness evaluation according to number of concrete services

The diagrams above (Fig. 4.13 and Fig. 4.14) demonstrate that the DBTLBO algorithm consistently outperforms the TLBO, GA, and PSO algorithms across multiple instances of service selection execution. The fitness values obtained by DBTLBO remain relatively stable even as the number of services (both concrete and abstract) increases. Additionally, the small standard deviation in fitness values, as shown in Tables 4.8 and 4.9, reflects the stability of the DBTLBO results.

In the third experiment, we assessed the convergence speed of the DBTLBO algorithm by comparing it with the TLBO, GA, and PSO algorithms. The test was conducted with varying numbers of concrete and abstract services. Fig. 4.15 illustrates an example of such a test, where the number of abstract services was fixed at 100, and the number of concrete services per abstract service was set to 300. It is observed that the fitness value for DBTLBO stabilizes at iteration 77, whereas the TLBO, PSO, and GA algorithms stabilize at iterations 269, 324, and 389, respectively.

As shown in the graph (Fig. 4.15), the DBTLBO algorithm exhibits faster convergence and achieves a higher fitness value compared to TLBO, GA, and PSO algorithms. This advantage is attributed to the local search in the Tutoring phase, where a subset of the learner's decision variables is replaced with the corresponding variables from the best solution found thus far. The size of this subset varies dynamically from one solution to another and is determined by the Alpha parameter, α, which is adjusted dynamically based on the algorithm's progress. This unique process sets DBTLBO apart from the competing algorithms, enabling it to quickly jump to the best-found solutions and thus avoid premature convergence, facilitating rapid convergence overall. Furthermore, the graph in Fig. 4.15 illustrates that DBTLBO initially explores the solution space and then rapidly converges to the best solution. The Restarting phase further demonstrates DBTLBO's ability to escape local optima and converge quickly to a superior solution.

4.4.4 Part 3 Conclusion

QoS-aware service composition is a critical step in dynamic business process formulation, enabling the creation of complex service compositions. As businesses increasingly rely on service-oriented architectures, the need for an efficient selection mechanism that ensures the required Quality of Service (QoS) is more important than ever. Given the vast number of available online services with functionally similar alternatives, selecting the optimal QoS-enabled web services has become an NP-hard problem. To address

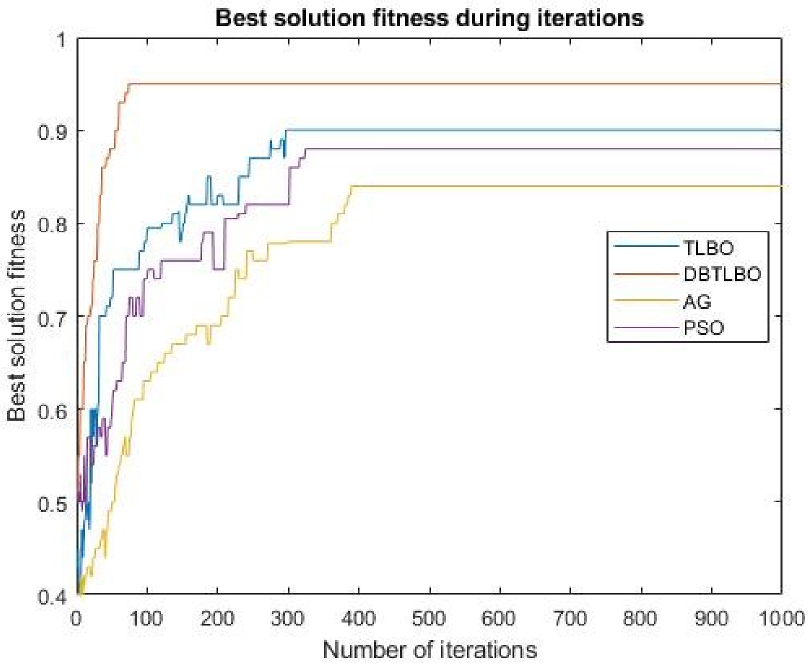

FIGURE 4.15: Evolution of the fitness value of the best solution during the iterations

this challenge, Meta-heuristic algorithms provide a promising approach for finding high-quality service compositions within a reasonable computational time.

In this part, we explored the potential of the BTLBO Meta-heuristic algorithm and developed its discrete version (DBTLBO), tailored specifically to the web service selection problem. The proposed approach focuses on selecting a specific composition of web services from a set of candidate services, optimizing the selection based on non-functional QoS properties such as response time, reliability, cost, reputation, and availability.

To evaluate the effectiveness of DBTLBO, several experiments were conducted using artificial service datasets to analyze how well the algorithm performs as the number of services increases. The results were compared with TLBO, GA, and PSO under the same execution environment. The findings indicate that DBTLBO outperforms the comparative algorithms, demonstrating superior performance in terms of fitness values and convergence speed.

The experimental results confirmed the strong convergence properties of DBTLBO, showing its ability to navigate complex problem spaces of varying sizes and consistently find optimal composition plans with maximum fitness values. These results encourage us to integrate the functional qualities of the services as well as the dependence constraints between the implementations of Web services, in the evaluation of quality of services, which remains a prospect for future work.

This approach offers a framework for creating and flexible e-government systems that can react to changes in real time. The DBTLBO-based approach makes significant improvements to maintaining service continuity, improving citizen satisfaction, and advancing the general quality and responsiveness of digital public services through enhanced scalability, reliability, and performance.

4.5 Conclusion

In this chapter, we presented our main contributions to the development of an integrated, user-centric, and interoperable e-government framework that is suited to the Algerian environment. These contributions revolve around three key components, each of which we discussed in detail in the previous sections. First, we proposed a conceptual model for processing citizen requests, utilizing Semantic Web technologies, for handling and successfully satisfy user needs. Second, we developed a citizen profile ontology that enhances the personalization and accessibility of e-government services through standardized and coherent representation of citizen data. Third, we addressed the challenge of QoS-aware service composition by introducing a discrete balanced teaching-learning optimization (DBTLBO) algorithm, aimed at optimizing service selection and composition based on non-functional QoS properties. These contributions combine to create a coherent and intelligent framework aimed at improving public service delivery and interoperability. They also lay the foundation for the creation of a prototype, presented in the next chapter that will facilitate data integration and exchange within the Algerian health information system.

Chapter 5

Use case study: Hospital Information System

5.1 Introduction

Interoperability in healthcare is a fundamental concept that allows various healthcare entities to exchange medical records effectively, ensuring coordinated patient care and continuous monitoring. In the context of Algeria, the healthcare sector offers a valuable opportunity to adopt modern IT systems that promote collaboration, facilitate seamless information exchange, and strengthen communication between hospitals, laboratories, and healthcare providers. These advancements are essential for delivering high-quality care while reducing healthcare costs and improving overall system efficiency.

This chapter provides an overview of the Algerian healthcare system and emphasizes the urgent need for a solution that enhances semantic interoperability among healthcare information systems. We explore various approaches specifically designed to tackle semantic interoperability challenges in healthcare. Additionally, we introduce our prototype, which was developed to enable the efficient exchange and sharing of health data within the Algerian context. Given the complexity of interoperability between diverse health information systems, our focus is placed on the business processes involved in data exchange between two hospitals.

5.2 Algeria's Health Status

After independence in 1962, the Algerian health system witnessed a transformation from an insufficient health system, suffering from disparities in the quality of services and concentrated only in the capital, Algiers, and other major northern cities, to a health system that works to improve coverage and quality of healthcare [200, 201]. Algeria government pays special attention to the health sector in terms of mobilized resources or adopted policies, with the aim of ensuring adequate care and its free provision in the public sector [202, 203].

Table 5.1 shows the positive impact of the state's social policy in the health sector, through the state's full financial support for health institutions, which has enabled an increase in the number of doctors [199]. However Table 5.2 illustrates the development of health institutions through the creation of new structures to respond to Algeria's current demographic growth [199].

Table 5.1: Number of Doctors in Algeria [199]

Individuals per Doctor	**Year**
7835	1962
4245	1976
1066	1994
640	2010
588	2018

Table 5.2: Health Institutions in Algeria [199]

Health Institution	**2000**	**2015**
University hospital institution	0	1
University hospital centers	15	12
Specialized hospital institutions	31	75
Specialized hospital institution maternal-childhood	0	30
Public institution for neighborhood health	497	1659
Treatment room	3964	5762

With the emergence of computer technologies, the Algerian government attaches particular interest to the digitalization of the health sector [202–204]: In this context, significant investments have been made, particularly in Information and Communication Technologies (ICT), leading to the launch of several e-health initiatives aimed at modernizing healthcare infrastructure, improving service delivery, and streamlining administrative processes. This digital shift holds the potential to greatly enhance the responsiveness and resilience of the Algerian healthcare system, especially in times of crisis. For example, during the COVID-19 pandemic, digital tools played a crucial role in enabling real-time data analysis, supporting decision-making, and optimizing the allocation and distribution of medical resources.

The development of digitization in Algeria's health sector is shown in Table 5.3[202]:

Table 5.3: Digital Health Transformation Timeline in Algeria

Period	**Stage**	**Key Developments**
2000 – 2010	Initial Phase: Foundations	Introduction of basic hospital information systems (e.g., appointment scheduling, billing, and electronic record-keeping). Systems were limited and lacked integration.
2011 – 2015	Transition to National Strategy	Deployment of EHRs and Health Information Systems (HIS) in major hospitals like those in Algiers and Oran. However, progress was hindered by infrastructure gaps and resistance to change.
2016 – 2019	Expansion of Digital Initiatives	Strengthening of digital health efforts through the Electronic Medical Record project. Development of telemedicine programs to enhance access to care in remote areas.
2020 – 2021	Response to COVID-19 Pandemic	Rapid digital transformation triggered by the pandemic. Emergence of telehealth platforms, smartphone-based consultations, and electronic health portals. Enhanced health data management to track COVID-19 cases.
2022 – Present	Acceleration of Digital Transformation Strategy	Continued development of digital health infrastructure with the integration of AI, IoT, and big data analytics. Wider adoption of electronic medical records (EMRs) across health institutions, enabling centralized access to patient information.

Despite this commitment to accessible healthcare, significant challenges remain, particularly in the adoption and integration of modern IT solutions within the medical sector.

The lack of digital infrastructure in many healthcare institutions presents major barriers to efficiency and service delivery. Several critical issues include:

- Lack of Centralized Medical Records: Patient medical histories, including consultations, diagnostic tests, and surgical procedures, are often not properly recorded or shared between healthcare institutions. This lack of information delays treatment and, in critical cases, may even prevent patients from receiving appropriate medical care.
- Limited System Interconnectivity: Many healthcare providers operate in isolation, with incompatible information systems that cannot communicate with one another. This prevents seamless data exchange, making remote collaboration between hospitals and clinics difficult.
- Impact on Healthcare Planning: The absence of structured health data hinders effective decision-making, including service allocation, national screening program implementation, and resource distribution. Without accurate data, policymakers struggle to optimize healthcare investments and improve public health outcomes.

Efforts to integrate healthcare information systems have been initiated to improve data sharing and collaboration among hospitals, laboratories, and healthcare providers. However, several challenges persist, including:

- Data Security and Privacy Risks: Ensuring patient confidentiality while enabling data sharing remains a critical concern.
- Technical Barriers in Data Transmission: Existing IT infrastructure lacks standardized communication protocols, complicating data exchange.
- Network Limitations: Many healthcare institutions, particularly in rural or underserved areas, lack reliable digital connectivity, further restricting integration efforts.

Among these challenges, system and data interoperability stands out as the most significant barrier to achieving a fully integrated and efficient healthcare system. Overcoming this issue will require robust digital solutions, such as ontology-based frameworks and semantic web technologies, to ensure seamless communication and interoperability between different healthcare entities.

Addressing these challenges is crucial for Algeria to modernize its healthcare infrastructure, optimize resource allocation, and ultimately improve patient care and public health outcomes.

5.3 Interoperability in Healthcare

Interoperability refers to the sharing and use of information or data by two or more systems or applications [205]. In healthcare, interoperability includes the seamless and secure exchange of electronic health information across different systems and environments [206–208], which helps:

- Enabling healthcare providers to access and share patient data efficiently.
- Coordinating care for chronic patients by enabling collaboration between different eHealth systems, such as electronic health records (EHRs), personal health records (PHRs), and wireless medical sensors.
- Facilitating clinical research by sharing EHRs among healthcare providers.
- Facilitates more informed decision-making, which improves the overall quality of healthcare services.
- Delivering high-quality healthcare and reducing healthcare costs.

According to [209, 210], interoperability can be classified into three categories:

- Fundamental Level: describes the prerequisites for securely connecting different systems and exchanging data.
- Structural Level: defines the format, syntax, and data that needs to be interpreted at the field level.
- Semantic level: Refers to the interoperability of processes enabling full interpretation of understanding and the use of publicly defined terminologies, vocabularies, and standardized values.

5.4 Electronic Health Records

Electronic Health Records (EHRs) have become a major objective of modern health systems, which have been largely adopted by hospitals and institutions to improve patient care and rationalize clinical workflow. It is a unified digital platform that often includes data from various health systems, offering health professionals and full access to patient information. This integration improves the accuracy of the diagnosis, supports decision -based decision -making and helps improve treatment results. In addition, EHRs allow a transparent exchange of information on different systems, improving the coordination and continuity of improved care.

The EHR can be considered the natural evolution of the physical medical record, which consisted of paper medical documents for a patient. As a digital representative of medical records in traditional paper documents, the EHR includes administrative and clinical data for patients, in particular, the demographic data, medical history, prescriptions, laboratory results, allergies, vital signs, radiology reports, problem lists, and progress notes [211]. Figure 5.1 shows the main parts participating in the EHR environment.

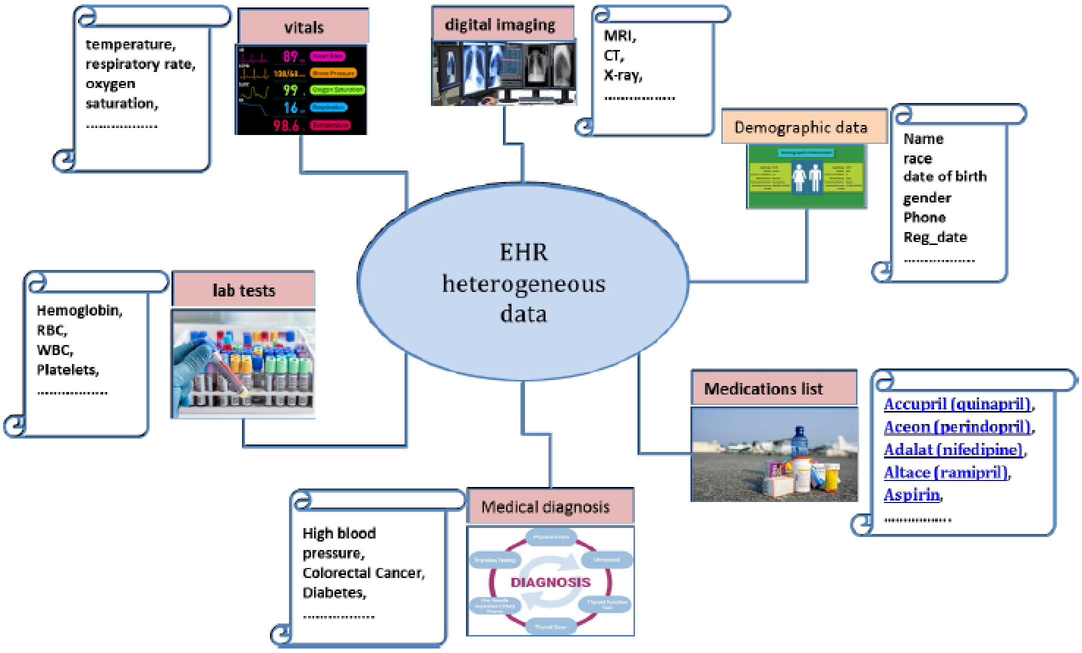

FIGURE 5.1: EHR heterogenneous data [211]

The electronic health record (EHR) can function more efficiently if the syntax and semantics of the data are implemented in an interoperable manner; By ensuring a common understanding of the medical data being managed, it reduces the risk of medical errors.

It also efficiently and quickly organizes the collaboration of different healthcare providers (hospitals, government agencies, clinics, general practitioners, etc.) [213]. However, to achieve an electronic medical record, a number of obstacles must be overcome [214], such as understanding a wide range of terminology, resolving ambiguities, and defining and updating concepts. In [215], the authors discuss the interoperability of electronic health record from a management and business standpoint, explaining the links between healthcare interoperability and other areas such as telemedicine, big data, and strategic surveillance.

To accomplish good semantic interoperability between EHRs by establishing a shared vocabulary and conceptual understanding of clinical data, three key components are required: EHR standards, terminologies, and ontologies [216]. A description of the most used of these elements is presented in Table 5.4.

Table 5.4: Components of Semantic Healthcare Interoperability

Standard	**Description**	**Website**
HL7 (Health Level Seven)	ANSI-accredited standards for electronic data exchange in healthcare, focusing on the Application layer of the OSI model.	www.hl7.org
HL7 Sub-standards	HL7 CCR (Continuity of Care Record) HL7 CDA (Clinical Document Architecture) HL7 HQMF (Health Quality Measure) HL7 CDD (Continuity of Care Document) HL7 V2 HL7 V3 (Reference Information Model - RIM) HL7 VMR (Virtual Medical Record)	www.hl7.org
CEN/ISO EN 13606	European/ISO standard for EHR interoperability, standardizing the structure and exchange of health information.	www.iso.org
DICOM	Universal standard for medical imaging and related information exchange.	www.dicomstandard.org
ICD (International Classification of Diseases)	WHO-maintained system for categorizing diseases and health conditions.	www.who.int
LOINC	Standardized identifiers for health measurements, observations, and documents.	loinc.org
SNOMED CT	Clinical terminology for standardized representation of medical phrases.	www.snomed.org
OWL (Web Ontology Language)	Markup language for exchanging ontologies, supporting the Semantic Web in healthcare.	www.w3.org

5.4.1 EHR Standards

EHR standards define the architectures that enable EHR systems from different developers to communicate and exchange data efficiently [212]. By promoting interoperability, these standards play a vital role in improving the quality of care, ensuring that relevant health information is accessible to the right people at the right time. Several international standards bodies (SDOs), including the International Organization for Standardization (ISO), the European Committee for Standardization (CEN), and Health Level Seven (HL7), have introduced widely adopted interoperability standards aimed at enabling the seamless exchange of health information between diverse healthcare environments [211]. However, despite the development of multiple EHR interoperability standards, there is no single universal standard, and complete interoperability between systems remains an ongoing challenge. Table 5.5 presents a comparison of the most commonly used health standards.

5.4.2 Medical terminology for EHRs

Medical terminology is a collection of related concepts and words that provide structured medical expressions and a controlled vocabulary to support clinical care, decision-making, search engine optimization, and quality improvement [217]. Terminologies may include clinical reference terminologies or interface terminologies. Clinical reference terminologies is a collection of terms, concepts, and relationships, in human language that have a semantic hierarchy and serve as a common reference point for comparing and aggregating medical data. Examples of reference terminologies:ICD-9-CM, LOINC, SNOMED CT. Interface terminology, also referred to as input terminology, application terminology, or colloquial terminology, is a systematic set of healthcare-related terms that aid clinicians in entering patient data into computer systems. Examples of interface terminology: NANDA, local expressions, and preferences.

5.4.3 Ontology for EHRs

In the context of knowledge representation and information science, ontology is the formal and explicit specification of a common conceptualization within a particular field. It is basically a structured framework that outlines the entities, concepts, and

TABLE 5.5: Comparison of Healthcare Interoperability Standards

Standard	**Data Type**	**Interoperability**	**Primary Role**	**Focus Area**	**Used For**
HL7 (v2/v3)	Structured messages, clinical & admin data	Syntactic + partial Semantic	Clinical messaging and data exchange	Communication across clinical systems	Patient records, lab results, ADT (Admission/Discharge/-Transfer)
CEN/ISO EN 13606	Structured EHR content (dual-model)	Semantic	Standardized EHR data sharing	EHR structure and semantic consistency	Longitudinal records, cross-organization EHR exchange
DICOM	Medical images + metadata	Technical + Syntactic	Image communication & integration	Radiology and imaging systems	Imaging data sharing (e.g., CT, MRI)
ICD	Disease & condition codes (numeric)	Semantic	Diagnostic classification & reporting	Diagnoses, mortality, epidemiology	Billing, public health stats, diagnosis coding
LOINC	Observation & measurement codes	Semantic	Lab and clinical observation coding	Lab tests, vital signs, documents	Standardizing lab results, data sharing, clinical aggregation
SNOMED CT	Detailed clinical terminology	Rich Semantic	Clinical concept representation	Findings, procedures, anatomy	EHR entry, decision support, analytics, documentation

relationships pertinent to a given field of study. By capturing and representing the underlying semantics of information, ontologies help people and systems communicate by establishing a common vocabulary and understanding [218, 219]. As a central component of the Semantic Web, ontology can be created using any computer description language, including the Web Ontology Language (OWL) and the Resource Description Framework (RDF).

Since ontologies provide consistent semantics and explicit formal models, several works expressed interest in creating distributed and interoperable electronic health record environments based on ontologies. [212] proposed a semantic ontology-based model that can unify different health record environments (EHRs) data formats. [220] developed a method of semi-automated integration to convert different classical data sources (like RDB, XML, and UML) into local ontologies (OWL2). In order to identify related concepts and avoid their duplication in the combined output, these ontologies were then integrated into a global ontological model using syntactic, structural, and semantic similarity measurement techniques. A two-step fusion of electronic medical record data with ontologies is proposed by [221] In the first step, specific data and keywords from various EHR formats are converted into a more easily parsable format (JSON format). In the second step, the obtained JSON data is combined with specialized medical ontologies. Several medical data standards for heterogeneous EHRs were used to implement the multi-step semantic architecture proposed by Kiourtis [222]. The suggested framework included a mechanism for removing explicit space data from ordered EHR datasets and converting them to Common Health Language via ontologies. Furthermore, the use of ontologies has shown promise in mapping scenarios for data access rules. The authors in [223] employ an ontology-based mediation service to enhance the physician's query and utilizes the MiniCon algorithm for the creation of a composite web service. In [224], an agent coordination infrastructure maps community organizations' access rules to the data using an OWL (Web Ontology Language) ontology.

5.5 Scenario: The Need for Interoperability in Emergency Healthcare

In this scenario, a patient arrives at the medical-surgical emergency department of Hospital A in critical condition. However, his complete medical history, including surgical reports, treatment protocols, imaging tests, and biological evaluations, is stored at Hospital B, where he was previously hospitalized.

At the time of admission, the patient does not have access to his medical records, which are essential for accurate diagnosis and effective treatment. Given the severity of his condition, the attending physicians at Hospital A require immediate access to his past medical data to make informed clinical decisions. However, the lack of an interoperable system prevents real-time communication between the two hospitals, leading to delays in treatment and potential medical risks.

This scenario highlights the critical need for interoperability in healthcare, emphasizing the importance of seamless data exchange between medical institutions. Implementing ontology-based semantic interoperability would enable instant retrieval of patient records, ensuring timely, accurate, and coordinated medical care across different healthcare facilities.

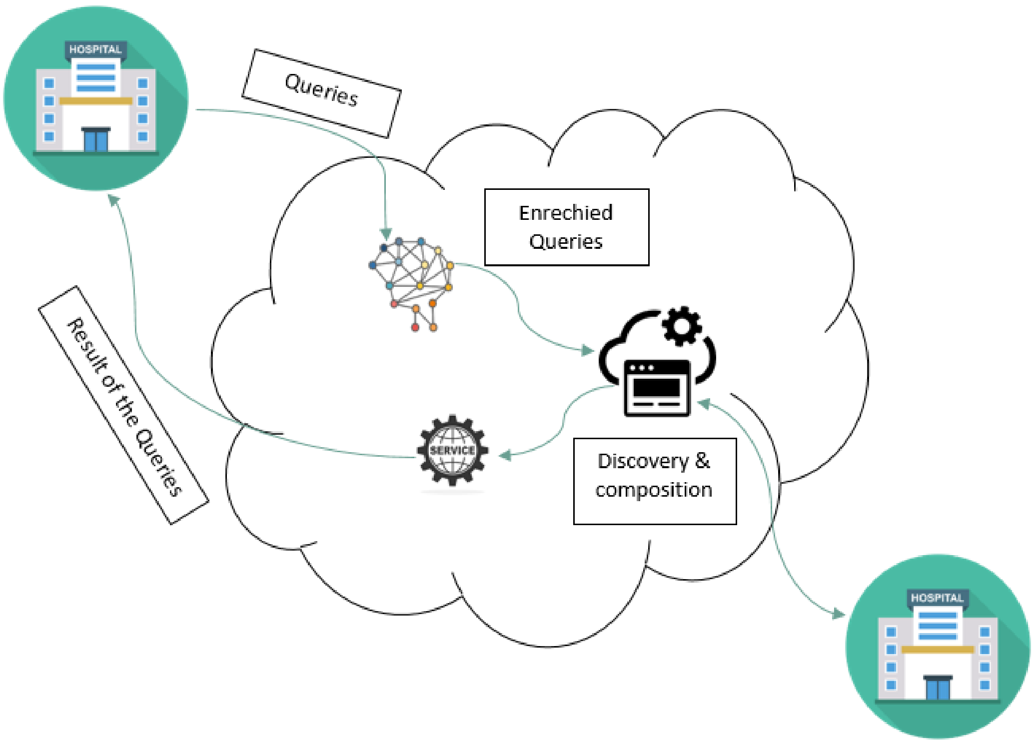

FIGURE 5.2: Scenario of exchange data between two hospitals

5.6 The Proposed Model

In our proposed model, hospitals are interconnected through a semantic interoperability framework built on semantic web service composition, ensuring seamless data exchange between healthcare institutions. Each patient is assigned a unique identifier, which facilitates accurate medical record retrieval and synchronization across different hospitals. The model is structured around a set of ontologies that define both domain-specific medical concepts and service functionalities, enhancing interoperability and knowledge representation.

Web services play a crucial role in modern software architectures, enabling efficient communication and data exchange between heterogeneous applications and systems over the internet. By promoting interoperability, web services allow diverse software platforms to interact, regardless of their underlying technologies, data formats, or communication protocols.

Web service descriptions are generally categorized into two approaches:

- Semantic Description Languages – Derived from the semantic web, these languages enhance machine-readable service descriptions, enabling advanced reasoning and automated service discovery.
- Annotation-Based Descriptions – These leverage the extensibility features of existing web service languages by incorporating semantic annotations to enrich service descriptions.

In our model, we have opted for the annotation-based approach, where each Health Information System (HIS) application function is semantically annotated using a local ontology. This approach ensures that all web services within the hospital network are uniformly described, allowing for consistent integration and interaction across different medical institutions.

Considering that all hospitals utilize a standardized set of web services, our model includes the following key web services commonly used in Health Information Systems (HIS):

Table 5.6: Medical Web Services and Their Functionalities

Service	**Functionality**
`Ser_patient`	Allows for the exchange of patient data
`Ser_examination`	Provides access to the different examinations of a patient
`Ser_consultation`	Allows retrieval of various consultations of a patient
`Ser_history`	Returns information about a patient's medical history
`Ser_result`	Enables the exchange and retrieval of all patient test results

This model ensures that hospitals can efficiently share patient data, improving diagnostic accuracy, treatment efficiency, and overall healthcare coordination through semantic interoperability and web service integration.

Web services enable users to execute queries and retrieve relevant information. However, when dealing with complex queries that require data from multiple sources, web service composition becomes essential. The process of identifying and integrating suitable web services is known as Web Service Discovery.

Web service composition involves creating new services by integrating multiple existing and published services, thereby enhancing their overall functionality and value. It is classified into two main categories:

- Static Web Service Composition – Requires the manual selection of individual web services and their execution order before processing the request. This approach is predefined and lacks flexibility.
- Dynamic Web Service Composition – Determines the optimal combination of web services during the execution process, allowing for real-time adaptation based on the user's request.

In our work, we focus on dynamic web service composition, which offers greater flexibility and efficiency in processing complex healthcare queries. Several existing dynamic composition approaches have been explored in prior research, emphasizing the need for intelligent service selection and integration.

Our interoperability model is designed to support the dynamic composition of web services for handling complex healthcare queries. It is built on a mediator architecture, which utilizes:

- A global ontology to structure and unify medical data.
- Semantic web services to enable seamless communication between different healthcare information systems.

The process of executing a complex healthcare query follows these key steps:

- A healthcare professional submits a query in natural language to search for patient information.
- The system interprets and enriches the query using semantic terms derived from the global ontology.

- The necessary web services required to process the enriched query are discovered.
- The identified web services are then dynamically composed and executed.
- The requested data is retrieved and delivered to the healthcare professional.

By leveraging semantic enrichment, ontology-driven discovery, and dynamic composition, our proposed model ensures accurate, efficient, and seamless data retrieval across heterogeneous healthcare systems. This approach significantly enhances interoperability, decision-making, and patient care in health information systems.

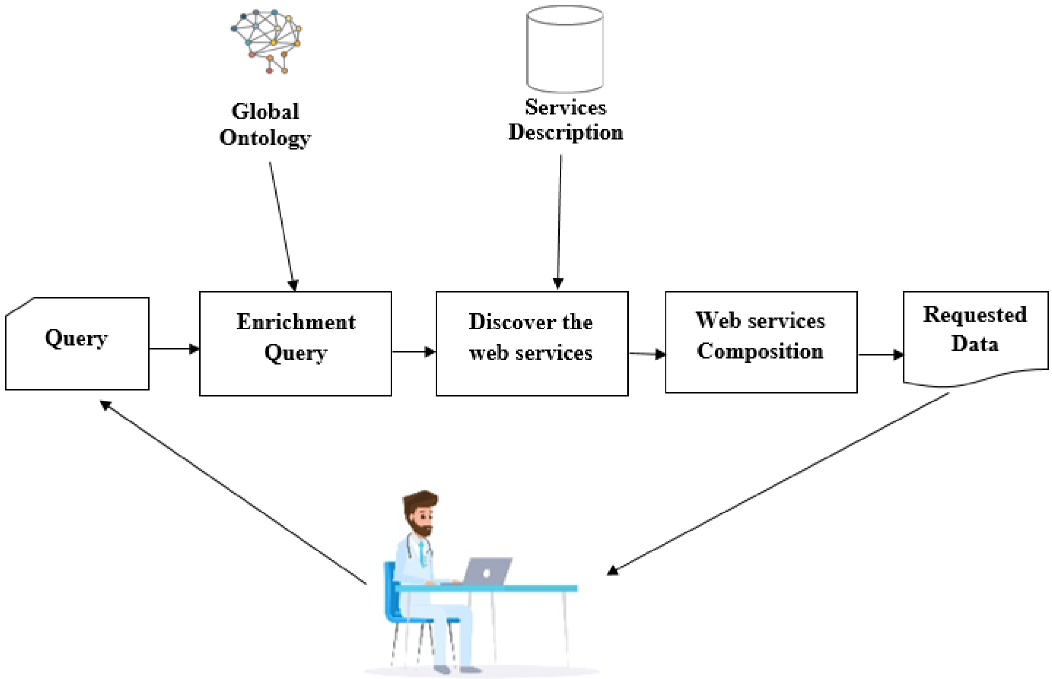

FIGURE 5.3: Architecture of the proposed model for HIS interoperability

5.7 The Global Ontology

Our proposed model employs a mediator-based architecture that utilizes a global ontology to facilitate semantic interoperability between healthcare information systems (HIS). This global ontology is derived from local ontologies, which are generated from individual HIS applications. By integrating these local ontologies, the system ensures a standardized representation of medical data, enabling seamless communication across different hospitals and healthcare providers.

The creation of the global ontology involved collaborations with healthcare professionals, including a Chirurgical Doctor from the CHU of Batna, Algeria. During discussions, she provided insights into common medical terminologies and concepts used across hospitals, doctors, and other medical staff. This expert knowledge was instrumental in identifying key terms, medical classifications, and relationships that needed to be included in the ontology.

Additionally, we merged multiple local ontologies derived from HIS databases. These local ontologies were constructed based on existing HIS data structures, ensuring that the global ontology accurately reflects real-world medical data exchanges. The result is a comprehensive and unified global ontology that serves as the foundation for semantic interoperability within our HIS model.

By employing this ontology-driven approach, our model enhances data standardization, consistency, and efficient healthcare information exchange, ultimately improving collaboration and decision-making in medical environments.

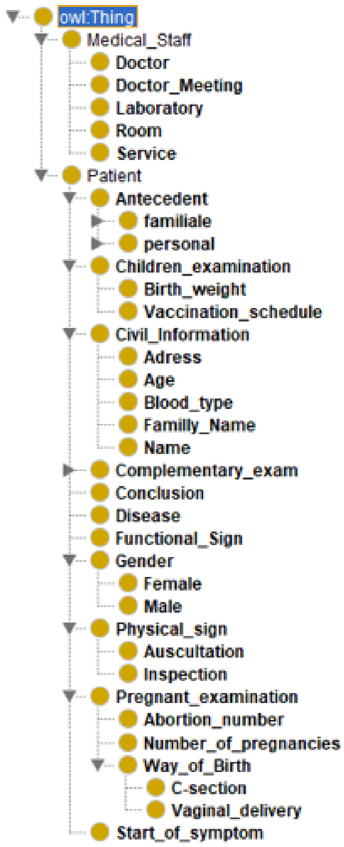

FIGURE 5.4: Class hierarchy for HIS model

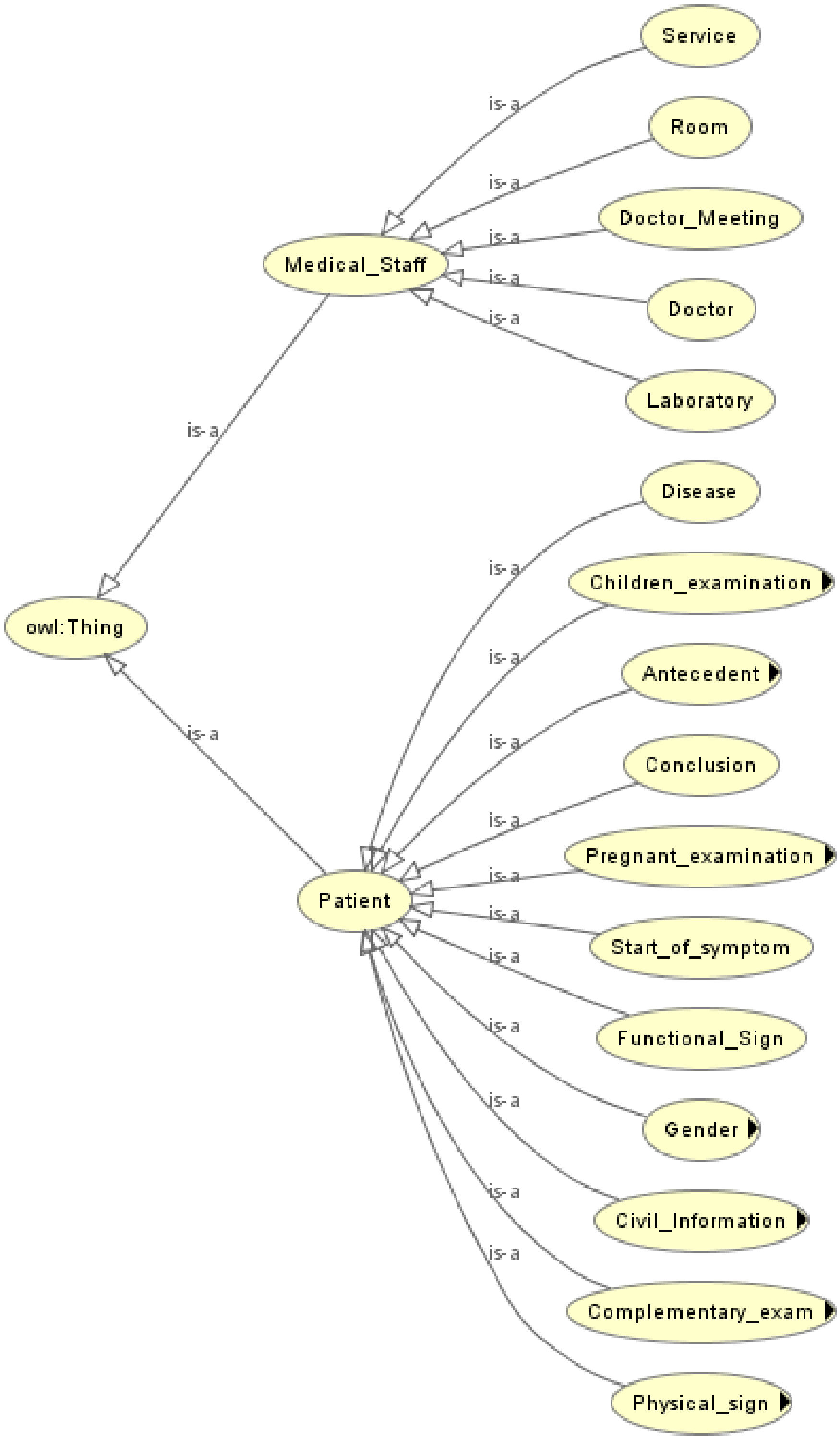

FIGURE 5.5: The global ontology for HIS model

5.8 Conclusion

Healthcare is a universal fundamental right, recognized in all Algerian constitutions since independence. This work examines the healthcare landscape in Algeria and emphasizes the need for improved semantic interoperability between health information systems (HIS).

Given the complexity of data exchange in healthcare, we focused on a business process scenario involving two hospitals. In this scenario, hospital information systems (HIS) communicate via web services in response to queries from healthcare staff. Since complex queries require multiple integrated web services, web service composition becomes essential.

Our proposed interoperability model facilitates web service composition for complex queries using a mediator-based global ontology and semantic web services. This global ontology, developed after extensive consultation with surgical specialists at CHU Batna, Algeria, integrates local ontologies derived from HIS applications. The model ensures efficient communication and seamless data exchange between interconnected HIS components, while also being scalable and adaptable for future expansions.

General Conclusion

In this thesis, we have addressed the interoperability issues in e-government platforms. We have presented a critical analysis of e-government in Algeria, which reveals both the progress made in modernizing government public information systems and administrative processes services and the persistent challenges encountered in implementing these systems. The study emphasized the obstacles confronting Algeria's e-government system, such as fragmentation, insufficient integration, and ineffective service delivery. To address Algeria's e-government systems requirements, we have explored the semantic web technologies and the composition of web services that promise to create an easy-to-use, interoperable, and integrated e-government platform. Applying ontologies and Semantic Web Services (SWS), the research has contributed:

- A citizen-centric web services integration modeling framework that aims to improve the integration and delivery of e-government services while enhancing interoperability between government agencies, standardizing knowledge representation, and optimizing web service configuration. By providing a unified methodological approach, this framework will significantly improve the efficiency and success of future e-government service integration and delivery projects.

- A citizen profile ontology improves e-government services in Algeria and supports decentralized administrative processes available to citizens. Using data processing and services across several government agencies, this ontology guarantees interoperability between different e-government systems and fosters a shared knowledge awareness. Improving the effectiveness and efficiency of services greatly increases the value that finally helps people. Moreover, this ontology may be expanded to cover new e-government services for people and companies, enabling the integration of present data across several platforms.

- An effective web service composition using an improved BTLBO algorithm enables the selection of an optimal composition from a set of candidate services, considering non-functional quality of service properties. This approach enables e-government systems to respond to real-time changes in a dynamic environment, providing dependable, scalable, cost-effective, secure, and high-performance services. By incorporating these capabilities, e-government systems can maintain service continuity, enhance user satisfaction, and improve service delivery, even in unpredictable or fluctuating conditions.

- A prototype facilitating the exchange and sharing of data and information in the health information system in Algeria. Given the complexity of interoperability between different health information systems, we focused on the business process involved in data exchange between two hospitals. We created a scenario in which the hospital information systems of the two hospitals exchange data via web services in response to a request from healthcare staff. The proposed interoperability model enables the composition of web services to handle complex queries using a global ontology-based mediator and semantic web services.

The results of this research will strongly contribute to establish digital transformation in Algeria by supporting the formation of a more cohesive e-government system capable of serving citizens more effectively. Some potential future directions with great promise could further improve these contributions, such as effectiveness, efficiency, and accessibility of e-government services. Additionally, responsiveness, security, and the capability to meet citizens' changing demands in a dynamic technology environment will also be thrilling improvements in the future. The following is a summary of the study's future directions:

- Although this thesis proposes a conceptual model to support the interoperability of e-government platforms, future research could address the practical challenges of implementing these models in cloud-based e-government systems. It would be relevant to examine how cloud computing can improve the flexibility, cost-effectiveness, and accessibility of e-government services while ensuring compliance with national and international regulations.

- As a future proposal to improve the performance of this framework, it is necessary to expand the scope of the citizen ontology to include other services. We also note the need to integrate other ontologies that touch on various institutions and administrations to ensure interaction between them. These improvements will not only ensure interaction between institutions and administrations to meet citizen needs but will also extend to meet the needs of the institutions themselves. Furthermore, expanding the scope of the ontology will enhance rapid adaptation to available technologies and the development of e-government services.

- Finally, hybridizing our proposed work with learning techniques and algorithms for predicting and dynamically adjusting the composition of web services is a promising improvement that will enable us to support the reliability and expand the performance of e-government services and cope with fluctuations in demand and conditions.

Bibliography

[1] Grigalashvili, V. (2022). E-government and E-overnance: Various or multifarious concepts. International Journal of Scientific and Management Research, 5(1). https://doi.org/10.37502/IJSMR.2022.5111

[2] Goloshchapova, T., et al (2023). E-government as a key to the economic prosperity and sustainable development in the post-COVID era. Economies, 11(4), 112. https://doi.org/10.3390/economies11040112

[3] Bensaoula, S., & Bezzar, M. S. (2025). E-governance in Algeria: Digital evolution and the challenges of public networks. Revue Algérienne de Finances Publiques, 15(1), 139–150.

[4] Idoughi, D. (2013). Algerian e-gov strategy and achievements. International Journal of eBusiness and eGovernment Studies, 5(1).

[5] Sidi Mammar, L., & Si Amer, H. (2025). Digitalization of the Algerian economy post-COVID-19: State of play and challenges. Al-Riyada for Business Economics Journal, 11(1).

[6] Abderahman, R., et al. (2022). Charting past, present, and future research in the Semantic Web and interoperability. Future Internet, 14(6), 161. https://-doi.org/10.3390/fi14060161

[7] Buranarach, M., et al (2018). Ontology-based knowledge management in government: Thai Excise Department. IEICE Transactions on Information and Systems, E101-D(4), 1105–1115.

[8] Sapel, P., et al (2024). Review and classification of manufacturing ontologies. Journal of Intelligent Manufacturing. https://doi.org/10.1007/s10845-024-02425-z

[9] Vitvar, T., et al (2010). Semantic technologies for e-government: An overview. Springer.

[10] Gabrel, V. (2015). Web services composition: Complexity and models. Discrete Applied Mathematics, 196, 100–114. https://doi.org/10.1016/j.dam.2014.07.010

[11] World Bank. (2002). E-government and governance: Lessons for Argentina. https://documents1.worldbank.org

[12] Organisation for Economic Co-operation and Development (OECD). (2003). The e-government imperative. OECD Publishing. https://-doi.org/10.1787/9789264101197-en

[13] United Nations Department of Economic and Social Affairs (UNDESA). (2022). The future of digital government. In UN E-Government Survey. United Nations. https://publicadministration.un.org/egovkb

[14] United Nations Department of Economic and Social Affairs (UNDESA). (2008). United Nations e-government survey 2008: From e-government to connected governance. United Nations. https://publicadministration.un.org/egovkb/en-us/Reports/UN-E-Government-Survey-2008

[15] United Nations. (2008). Global e-government survey 2008. United Nations. http://unpan1.un.org/

[16] Bhatnagar, S. (2002). E-government lessons from implementation in developing countries. Regional Development Dialogue, 23(2), 1–12.

[17] Gichoya, D. (2005). Factors affecting the successful implementation of ICT projects in government. Electronic Journal of e-Government, 3(4), 175–184.

[18] Batmetan, J. R., et al. (2022). IT infrastructure library framework for e-government maturity. International Journal of Information Technology Education.

[19] Setyawan, A. C. (2024). Enhancing public service delivery. Global International Journal of Innovative Research, 2(10), pages 2439-2453.

[20] Tennakoon, W. (2020). E-governance way forward: Challenges and opportunities for developing countries. Evidences from Sri Lanka. International Journal of Business, Economics and Law, 21(2). ISSN 2289-1552.

[21] Malodia, S., et al (2021). Future of e-government: Conceptual framework. Technological Forecasting and Social Change, 173, Article 121035. https://-doi.org/10.1016/j.techfore.2021.121035

[22] Widiamoko, C., et al. (2023). E-government and public participation. ARRUS Journal of Social Sciences and Humanities, 3(3).

[23] Patnaik, P., et al (2020). Use of data analytics for effective e-governance: A case study of "EMutation" system of Odisha. In S. Borah & E. Balas (Eds.),

[24] Pham, L., et al (2023). E-government service quality, perceived value, satisfaction, and loyalty: Evidence from a newly emerging country. Journal of Public Policy, 43(4), 812–833. https://doi.org/10.1017/S0143814X23000296

[25] Aritonang, D. M. (2017). E-government impact on public service quality. European Scientific Journal, 13(35), 99–112. https://doi.org/10.19044/esj.2017.v13n35p99

[26] Bhuiyan, S. H. (2011). Modernizing public administration in Bangladesh: Toward a comprehensive reform. Government Information Quarterly, 28(1), 54–65. https://-doi.org/10.1016/j.giq.2010.06.010

[27] Rossel, P., & Finger, M. (2007). Conceptualizing e-governance. In Proceedings of the 1st International Conference on Theory and Practice of Electronic Governance (ICEGOV 2007) (pp. 399–407). ACM. https://doi.org/10.1145/1328057.1328143

[28] Bertot, J. et al (2010). ICTs for transparency and accountability in e-government and social media. Government Information Quarterly, 27(3), 264–271. https://-doi.org/10.1016/j.giq.2010.03.001

[29] Deepak, D., & Saji, M. (2015). ICT infrastructure and e-governance performance. Springer.

[30] Alkhwaldi, A.,et al (2018). Cloud-based e-government services acceptance in Jordan. Journal of Information Technology and Software Engineering, 6(2), 1–8. https://-doi.org/10.4172/2165-7866.1000213

[31] Al-Omari, A., & Al-Omari, H. (2006). E-government readiness model. Journal of Computer Science, 2(11), 841–845. https://doi.org/10.3844/jcssp.2006.841.845

[32] Brown, H. D., & Thompson, S. (2011). ICT diffusion in Jamaica: A supply-side perspective. European Journal of Information Systems, 20(3), 293–304. https://-doi.org/10.1057/ejis.2011.6

[33] Layne, K., & Lee, J. (2001). Developing fully functional e-government: A four stage model. Government Information Quarterly, 18(2), 122–136. https://-doi.org/10.1016/S0740-624X(01)00066-1

[34] Chango, M. (2007). Challenges to e-government in Sub-Saharan Africa. In Proceedings of the International Conference on Electronic Governance (ICEGOV 2007)

[35] Bonham, G., et al (2001). Transformational e-government: Political leadership. In Proceedings of the Pan-European International Relations Conference.

[36] Bourn, J. (2002). Better public services via e-gov. UK Audit Office.

[37] Saleh, Z. I., et al (2013). SOA-based e-gov framework for Jordan. International Journal of Information Engineering and Electronic Business, 5(3), 1–8.

[38] Gitau, J. K., & Mburu, S. (2016). SOA model for e-gov systems in Kenya. American Journal of Information Systems, 4(3),

[39] Abied, O., et al. (2022). Cloud adoption in e-gov: A literature review. Pertanika Journal of Science and Technology, 30(1)

[40] Buyya, R., et al (2009). Cloud computing and IT platforms. Future Generation Computer Systems, 25(6), 599-616. https://doi.org/10.1016/j.future.2008.12.001

[41] Prithi, S., et al. (2024). AI and ML for e-gov data. In Managing data in e-governance. IGI Global. https://doi.org/10.4018/978-1-6684-7697-0.ch002

[42] Paul, A., & Paul, V. (2014). Framework for interoperable e-gov in India. Proceedings of the National Conference on e-Governance (NCeG), India.

[43] Lnenicka, M., & Komarkova, J. (2019). Big data governance in public sector: Managing digital transformation. International Journal of Information Management, 46, 124-135. https://doi.org/10.1016/j.ijinfomgt.2018.12.017

[44] Bélanger, F., & Hiller, J. S. (2006). A framework for e-government: Privacy implications. Business Process Management Journal, 12(1), 48-60. https://-doi.org/10.1108/14637150610643751

[45] Anas, K., et al. (2023). Trust and use of e-gov services. International Journal of Data and Network Science, 7(2), 689-696. https://doi.org/10.5267/j.ijdns.2023.2.003

[46] Zhao, J. J., & Zhao, S. Y. (2010). Opportunities and threats: A security assessment of state government websites. Government Information Quarterly, 27(1), 49-56. https://doi.org/10.1016/j.giq.2009.07.004

[47] Alrubaiq, A., & Alharbi, T. (2024). Cybersecurity for e-gov in KSA. Journal of Cybersecurity and Privacy, 4(1), 41-58.

[48] Lee, G., et al (2005). Current practices of leading e-government countries. Communications of the ACM, 48(10), 99-104. https://doi.org/10.1145/1089107.1089111

[49] Riley, T. B. (2007). Strategies for e-government implementation. Journal of Business & Public Policy, 1(1), 1-21.

[50] Cook, M., et al(2002). Making a case for local e-government. Center for Technology in Government, University at Albany. https://www.ctg.albany.edu/publications/reports/making_a_case

[51] Ebrahim, Z. (2005). E-government adoption in Bahrain [Doctoral dissertation, Brunel University].

[52] Forman, M. (2002). E-government strategy: Simplified delivery of services to citizens [Strategic report]. U.S. Office of Management and Budget. https://www.whitehouse.gov/wp-content/uploads/2018/06/egovstrategy.pdf

[53] Bachir, T. (2022). Le e-gouvernement en Algérie selon l'enquête 2022 des Nations Unies [E-government in Algeria according to the 2022 UN survey]. D4D Access. https://d4daccess.eu/sites/default/files/knowledge_base_products_files/E-government%20in%20Algeria.pdf

[54] Maouche, S., & Guemide, B. (2021). E-gov in Algeria: Prospects and constraints. Revue Algérienne des Sciences Juridiques et Politiques, 58(5), 123-145.

[55] Goel, S., et al (2012). Infrastructure challenges in e-governance. IOSR Journal of Computer Engineering, 5(2), 34-42. https://doi.org/10.9790/0661-0523442

[56] Benmoussa, M. S., & Ibrahim, M. B. (2018). E-gov adoption in Algeria. Journal of Advanced Research in Business and Management Studies, 10(1), 1-15.

[57] Adnane, H., & Benlahcen, L. (2022). E-gov and local development in Algeria. Revue Droit, Société et Pouvoir, 11(1)

[58] Bouguetaia, S., & Mezouri, E. (2024). E-governance in the Algerian public sector. Journal of Contemporary Issues in Business and Government, 30(1), 1-18

[59] Elgohary, E. (2017). E-gov implementation in developing countries. International Journal of Computer Techniques, 16(1), 12-21.

[60] Ahmed, S. M. Y., & Twinomurinzi, H. (2019). Semantic interoperability in Africa. Proceedings of the 52nd Hawaii International Conference on System Sciences. https://hdl.handle.net/10125/59891

[61] Akbar, P., et al (2022). Trends in e-government interoperability: A systematic literature review. Journal of e-Democracy and Open Government, 14(2), 1-25. https://-doi.org/10.29379/jedem.v14i2.693

[62] Bouklia, M., & Bouarioua, R. (2018). The evaluation of e-government in Algeria between reality and prospects. International Journal of E-Planning Research, 1(2), 107-119. https://ijep.dz/index.php/IJEP/article/view/19

[63] Gupta, P., et al (2024). Trust, risk, and privacy in e-government use: A meta-analysis. Information Systems Frontiers. Advance online publication. https://-doi.org/10.1007/s10796-024-10497-8

[64] Pennington, C. (2007). Introduction to web services. In J. Cardoso (Ed.), Semantic web services: Theory, tools and applications (pp. 134-154). IGI Global. https://-doi.org/10.4018/978-1-59904-045-5.ch007

[65] Huhns, M. N., & Singh, M. P. (2005). Service-oriented computing: Key concepts and principles. IEEE Internet Computing, 9(1), 75-81. https://-doi.org/10.1109/MIC.2005.16

[66] Erl, T. (2008). SOA: Principles of service design (Vol. 1). Prentice Hall.

[67] O'Sullivan, J., et al (2002). What's in a service? Distributed and Parallel Databases, 12(2-3), 117-133. https://doi.org/10.1023/A:1016548206222

[68] Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (PHd thesis, University of California, Irvine). http://www.ics.uci.edu/ fielding/pubs/dissertation/top.htm

[69] Pautasso, C., et al (2008, April). RESTful web services vs ”big” web services: Making the right architectural decision. Proceedings of the 17th International World Wide Web Conference (pp. 805-814), Beijing, China. https://-doi.org/10.1145/1367497.1367606

[70] Ngu, A. H. H., et al (2010). Semantic-based mashup of composite applications. IEEE Transactions on Services Computing, 3(1), 2-15. https://-doi.org/10.1109/TSC.2010.3

[71] World Wide Web Consortium. (2004, February 11). Web services architecture. https://www.w3.org/TR/ws-arch/

[72] Dialani, V., et al (2002). Transparent fault tolerance for web services based architectures. In European Conference on Parallel Processing (pp. 889-898). Springer. https://doi.org/10.1007/3-540-45706-2_116

[73] Yang, B., & Qin, Z. (2010). Composing semantic web services with PDDL. Information Technology Journal, 9(1), 48-54. https://doi.org/10.3923/itj.2010.48.54

[74] Jayakumar, S.,et al (2014). Suitable QoS parameters survey for standard web services & web applications to understand their cloud deployability. International Journal of Computational Intelligence Systems, 7(1), 1-18. https://-doi.org/10.1080/18756891.2013.865430

[75] Sirin, E., et al (2003, January). Semi-automatic composition of web services using semantic descriptions. In Proceedings of the 1st Workshop on Web Services: Modeling, Architecture and Infrastructure (pp. 17-24).

[76] Yu, Q., et al (2008). Deploying and managing web services: Issues, solutions, and directions. The VLDB Journal, 17(3), 537-572. https://doi.org/10.1007/s00778-006-0020-3

[77] Vinoski, S. (2002). Web services interaction models: Current practice. IEEE Internet Computing, 6(3), 89-91. https://doi.org/10.1109/MIC.2002.1003136

[78] Geer, D. (2003). Taking steps to secure web services. Computer, 36(10), 14-16. https://doi.org/10.1109/MC.2003.1236471

[79] Lau, T. Y., et al (2008). Public service in the information age: A study of e-government in Taiwan. Media Asia, 35(3), 190-197. https://-doi.org/10.1080/01296612.2008.11726883

[80] Venkatraman, S. S. (2004). Web services - The next evolutionary stage of e-business. Journal of International Technology and Information Management, 13(1), 1-15. https://doi.org/10.58729/1941-6679.1157

[81] Iosup, A., et al (2011). Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Transactions on Parallel and Distributed Systems, 22(6), 931-945. https://doi.org/10.1109/TPDS.2011.66

[82] Alpern, M. (2020). Critical success factors for e-government web services (Doctoral study). Walden University. https://scholarworks.waldenu.edu/dissertations/8832

[83] Bojang, M. B. (2021). Critical factors influencing e-government adoption in The Gambia. Society & Sustainability, 3(1), 39-51

[84] Semantic web. In Wikipedia. Retrieved , from https://en.wikipedia.org/wiki/Semantic_Web

[85] World Wide Web Consortium. Semantic web activity homepage. Retrieved from https://www.w3.org/2001/sw/

[86] Berners-Lee, T., et al (2001). The semantic web. Scientific American, 284(5), 34-43. https://doi.org/10.1038/scientificamerican0501-34

[87] Dumbill, E. (2001). Building the semantic web. Knowledge Technologies Conference, Austin, TX, United States.

[88] Xiao, Y., et al (2007). An ontology for e-government knowledge modelling and interoperability. IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2007), Shanghai, China.

[89] Sabucedo, L. A., & Rifón, L. A. (2006). Semantic service oriented architectures for e-government platforms. American Association for Artificial Intelligence.

[90] Stojanovic, L., et al (2004, October). Managing changes in the ontology-based e-government. Proceedings of the 2004 CoopIS, DOA, and ODBASE Confederated International Conferences. Springer.

[91] Zemmouri, E., et al (2011). Intégration des connaissances de domaine dans un processus d'ECD multi-vues [Domain knowledge integration in a multi-view ECD process]. 4èmes Journées Francophones sur les Ontologies (JFO2011) (pp. 73-85).

[92] Charlet, J., et al (2004). Ontologies pour le Web sémantique [Ontologies for the semantic web]. Revue Information, Interaction, Intelligence (I3), 4(2)

[93] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199–220. https://doi.org/10.1006/knac.1993.1008

[94] Klein, M., et al (2000, August). The relation between ontologies and schema-languages: Translating OIL-specifications to XML-schema. In Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence. Berlin, Germany.

[95] Luke, S., & Heflin, J. (1997). SHOE: Proposed specification [Technical report]. University of Maryland. http://www.cs.umd.edu/projects/plus/SHOE/spec.html

[96] Noy, N. F., et al (2003). Protégé-2000: An open-source ontology-development and knowledge-acquisition environment. AMIA Annual Symposium Proceedings, 2003, 953. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1480006/

[97] Bechhofer, S.,et al (2001). OilEd: A reasonable ontology editor for the semantic web. In Joint German/Austrian Conference on Artificial Intelligence (pp. 396-408). Springer. https://doi.org/10.1007/3-540-44795-4_34

[98] Sure, Y., et al (2002). OntoEdit: Collaborative ontology development for the semantic web. In The Semantic Web — ISWC 2002 (pp. 221-235). Springer. https://-doi.org/10.1007/3-540-48005-6_19

[99] Van Der Vet, P. E., & Mars, N. J. (1998). Bottom-up construction of ontologies. IEEE Transactions on Knowledge and Data Engineering, 10(4), 513-526. https://-doi.org/10.1109/69.706059

[100] Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. The Knowledge Engineering Review, 11(2), 93-136. https://-doi.org/10.1017/S0269888900007797

[101] Fernández-López, M., et al (1997). Methontology: From ontological art towards ontological engineering. AAAI Spring Symposium on Ontological Engineering, 33-40.

[102] Gruninger, M., et al (1997). Using process requirements as the basis for the creation and evaluation of process ontologies for enterprise modeling. ACM SIGGROUP Bulletin Special Issue on Enterprise Modelling, 18(3), 52-55.

[103] Lozano-Tello, A., & Gómez-Pérez, A. (2004). ONTOMETRIC: A method to choose the appropriate ontology. Journal of Database Management, 15(2), 1-18. https://-doi.org/10.4018/jdm.2004040101

[104] Sedrani, A., & Touir, A. (2016). Web service composition processes: A comparative study. International Journal on Web Service Computing, 7(1)

[105] Hesami Rostami, N. (2013). Web services composition methods and techniques: A review. International Journal of Computer Science, Engineering and Information Technology, 3(6)

[106] Lemos, A. L., et al (2016). Web service composition: A survey of techniques and tools. ACM Computing Surveys, 48(3), Article 33. https://doi.org/10.1145/2831270

[107] Zeshan, F., & Mohamad, R. (2011). Semantic web service composition approaches: Overview and limitations. International Journal on New Computer Architectures and Their Applications (IJNCAA), 1(3), 640–651.

[108] Sharmila, M. G. S., & Samuel, S. J. (2014). A trend analysis on web service composition. International Journal of Innovative Research in Science, Engineering and Technology, 3

[109] Sheng, Q. Z.,et al (2014). Web services composition: A decade's overview. Information Sciences, 280, 218-238. https://doi.org/10.1016/j.ins.2014.04.054

[110] Furkh Zeshan. et al (2011). Semantic web service composition approaches: Overview and limitations. International Journal on New Computer Architectures and Their Applications, 1(3), 640-651.

[111] Dimka, K., & Frank, L. (2008). Service-oriented architecture - Overview of technologies and standards. it - Information Technology, 50(2), 83-85. https://-doi.org/10.1524/itit.2008.9049

[112] Karoui, E., et al (2022). An agent and web services-based e-government model validation. Electronic Government, an International Journal, 18(3)

[113] van der Aalst, W. M. P., et al(1994). Modelling and analysing workflow using a Petri-net based approach. In Proceedings of the Second Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms (pp. 31-50)

[114] Object Management Group. (2011). Business process model and notation (BPMN) [Standard]. http://www.bpmn.org/

[115] Bekkouche, A., et al (2017). QoS-aware optimal and automated semantic web service composition with user's constraints. Service Oriented Computing and Applications, 11(2), 183-201. https://doi.org/10.1007/s11761-017-0205-1

[116] Russell, S. J., & Norvig, P. (2009). Artificial intelligence: A modern approach (3rd ed.). Prentice Hall.

[117] Medjahed, B.,et al (2003). Composing web services on the semantic web. The VLDB Journal, 12(4), 333-351. https://doi.org/10.1007/s00778-003-0101-5

[118] McDermott, D., et al (1998). PDDL - The planning domain definition language (Technical Report). Yale University. https://www.csee.umbc.edu/courses/671/fall12/hw/hw6/pddl1.2.pdf

[119] Yang, B., & Qin, Z. (2010). Composing semantic web services with PDDL. Information Technology Journal, 9(1), 48-54. https://doi.org/10.3923/itj.2010.48.54

[120] Ghallab, M., et al (2004). Hierarchical task network planning. In Automated planning: Theory and practice (pp. 229-262). Elsevier.

[121] Song, Y.,et al(2018). QoS-aware service composition using HTN planner. In 2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2) (pp. 107-110). IEEE. https://doi.org/10.1109/SC2.2018.00023

[122] McIlraith, S. A., & Son, T. C. (2002). Adapting Golog for composition of semantic web services. KR, 2, 482-493.

[123] Sohrabi, S., et al (2009). Web service composition via the customization of Golog programs with user preferences. In Conceptual modeling: Foundations and applications (pp. 319-334). Springer

[124] Lamine, R. B., et al (2017). Graph planning-based composition for adaptable semantic web services. Procedia Computer Science, 112, 358-368. https://doi.org/10.1016/j.procs.2017.08.009

[125] Carey, M. J. (2008). SOA what? IEEE Computer, 41(3), 92-94. https://doi.org/10.1109/MC.2008.82

[126] Sathya, M., et al (2010). Evaluation of QoS based web-service selection techniques for service composition. International Journal of Software Engineering, 1(5), 73-90

[127] Tarawneh, H., et al (2022). An intelligent cloud service composition optimization using spider monkey and multistage forward search algorithms. Symmetry, 14(1), 82. https://doi.org/10.3390/sym14010082

[128] Sefati, S., & Halunga, S. (2022). A hybrid service selection and composition for cloud computing using the adaptive penalty function in genetic and artificial bee colony algorithm. Sensors, 22(13), 4873. https://doi.org/10.3390/s22134873

[129] Kashyap, N., et al (2020). A multi-objective optimization using NSGA II for service composition in IoT. Procedia Computer Science, 167, 1928-1933. https://doi.org/10.1016/j.procs.2020.03.206

[130] Sestino, A., et al (2020). Internet of Things and Big Data as enablers for business digitalization strategies. Technovation, 98, 102173. https://doi.org/10.1016/j.technovation.2020.102173

[131] Hosseinzadeh, M., et al (2020). A hybrid service selection and composition model for cloud-edge computing in the Internet of Things. IEEE Access, 8, 85939-85949. https://doi.org/10.1109/ACCESS.2020.2992262

[132] Zhao, X., & Li, R. (2019). Advances on QoS-aware web service selection and composition with nature-inspired computing. CAAI Transactions on Intelligence Technology, 4(3), 159-174. https://doi.org/10.1049/trit.2019.0038

[133] Merzoug, M., et al (2014). QoS-aware web service selection based on harmony search. Proceedings of the 4th International Symposium on ISKO-Maghreb: Concepts and Tools for Knowledge Management (ISKO-Maghreb), 1–6. https://doi.org/10.1109/ISKO-Maghreb.2014.7033465

[134] Thangaraj, P., & Balasubramanie, P. (2021). Metaheuristic QoS-based service composition for service computing. Journal of Ambient Intelligence and Humanized Computing, 12, 5619-5625. https://doi.org/10.1007/s12652-020-02095-8

[135] Siarry, P., & Michalewicz, Z. (Eds.). (2007). Advances in metaheuristics for hard optimization. Springer.

[136] Khelil, H., Brahimi, M. Toward an efficient web service composition based on an improved BTLBO algorithm. J Supercomput 80, 8592–8613 (2024). https://-doi.org/10.1007/s11227-023-05777-0

[137] Taheri, A.,et al (2021). An efficient Balanced Teaching-Learning-Based optimization algorithm with Individual restarting strategy for solving global optimization problems. Information Sciences, 576, 68-104. https://-doi.org/10.1016/j.ins.2021.06.064

[138] Amiri, M., et al (2013). QoS-based web service composition based on genetic algorithm. Journal of AI and Data Mining, 1(2)

[139] Arasteh, B., et al (2024). A quality-of-service aware composition-method for cloud service using discretized ant lion optimization algorithm. Knowledge and Information Systems, 66, 4199-4220. https://doi.org/10.1007/s10115-024-02086-7

[140] Babamir, S. M., & Zahiri, N. (2025). A pattern and summarization based optimization algorithm to QoS-aware web service composition selection. Journal of Artificial Intelligence and Data Mining, 13(2), 183-206

[141] Pattanaik, K. K., et al (2017). BAT and hybrid BAT meta-heuristic for quality of service-based web service selection. Journal of Intelligent Systems, 26(1), 123-137. https://doi.org/10.1515/jisys-2016-0098

[142] Zhao, C. Y., et al (2014). A hybrid algorithm combining ant colony algorithm and genetic algorithm for dynamic web service composition. The Open Cybernetics & Systemics Journal, 8, 146-154

[143] Dahan, F., & Alwabel, A. (2023). Artificial Bee Colony with Cuckoo Search for solving service composition. Intelligent Automation & Soft Computing, 35(3)

[144] Pawar, S., & Prasanth, Y. (2017). Multi-objective optimization model for QoS-enabled web service selection in service-based systems. New Review of Information Networking, 22(1), 34-53.

[145] Zhang, Q., et al (2024). An adaptive robust service composition and optimal selection method for cloud manufacturing based on the enhanced multi-objective artificial hummingbird algorithm. Expert Systems with Applications, 244, 122823. https://doi.org/10.1016/j.eswa.2023.122823

[146] Jalal, S., & Dharmendra, K. Y. (2021). A multiobjective discrete grey wolf optimization approach for transactional and QoS-driven web services composition. Applied Artificial Intelligence, 35(15), 1646-1684

[147] Khadka, R., & Sapkota, B. (2010). An evaluation of dynamic Web service composition approaches. In Proceedings of the 4th International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing (ACT4SOC 2010) (pp. 67-79). SciTePress.

[148] Yu, T., et al (2007). Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Transactions on the Web, 1(1)

[149] Zeng, L., et al (2004). QoS-aware middleware for web services composition. IEEE Transactions on Software Engineering, 30(5), 311-327. https://doi.org/10.1109/TSE.2004.11

[150] Pop, C. B., et al (2011). A tabu search optimization approach for semantic web service composition. In 2011 10th International Symposium on Parallel and Distributed Computing (pp. 274-277). IEEE. https://doi.org/10.1109/ISPDC.2011.41

[151] Liu, Q., et al (2008). Web services composition with QoS bound based on simulated annealing algorithm. Journal of Southeast University (English Edition), 24(3), 308-311

[152] Meng, S. Y., & Zhang, C. C. (2021). Composition of web services of multi-population adaptive genetic algorithm based on cosine improvement. Journal of Computer and Communications, 9(4), 109-119. https://doi.org/10.4236/jcc.2021.94008

[153] Zhang, T. (2014). QoS-aware web service selection based on particle swarm optimization. Journal of Networks, 9(3)

[154] Dahan, F., et al (2021). An enhanced ant colony optimization based algorithm to solve QoS-aware web service composition. IEEE Access, 9, 34098-34111. https://-doi.org/10.1109/ACCESS.2021.3058731

[155] Chandra, M., & Niyogi, R. (2019). Web service selection using modified artificial bee colony algorithm. IEEE Access, 7, https://doi.org/10.1109/ACCESS.2019.2945549

[156] Seghir, A., et al (2019). An interval-based multi-objective artificial bee colony algorithm for solving the web service composition under uncertain QoS. The Journal of Supercomputing, 75(8), 4526-4554. https://doi.org/10.1007/s11227-019-02762-4

[157] Matsuoka, H., & Moustafa, A. (2022). Multi-task deep reinforcement learning for IoT service selection. In Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3 (pp. 548-554). SCITEPRESS. https://doi.org/10.5220/0010997000003116

[158] Wang, H., et al (2010). Adaptive service composition based on reinforcement learning. In International Conference on Service-Oriented Computing (pp. 92-107). Springer. https://doi.org/10.1007/978-3-642-17358-5_7

[159] Wang, H., et al (2016). Automatic hierarchical reinforcement learning for efficient large-scale service composition. In 2016 IEEE International Conference on Web Services (ICWS) (pp. 57-64). IEEE. https://doi.org/10.1109/ICWS.2016.15

[160] Wang, H., et al (2020). Integrating recurrent neural networks and reinforcement learning for dynamic service composition. Future Generation Computer Systems, 107, 551-563. https://doi.org/10.1016/j.future.2020.02.030

[161] Huan, L., et al (2007). Automatic Semantic Web Service Composition via Agent Intention Execution in AgentSpeak. In *2007 IEEE/WIC/ACM International Conference on Web Intelligence*

[162] Wei, L., et al (2010). Multi-agent based QoS-aware Service Composition. In 2010 IEEE International Conference on Systems, Man and Cybernetics (pp. 3125-3132). IEEE. https://doi.org/10.1109/ICSMC.2010.5642476

[163] Zouad, S., & Boufaida, M. (2020). Using multi-agent microservices for a better dynamic composition of semantic web services. In Proceedings of the 4th International Conference on Advances in Artificial Intelligence (pp. 47-52). https://doi.org/10.1145/3441417.3441423

[164] Krishnaveni, A., et al (2019). A survey on nature-inspired computing (NIC): Algorithms and challenges. Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence, 19(3), 1-15

[165] Al Mudawi, N., et al (2020). Issues and challenges: Cloud computing e-government in developing countries. International Journal of Advanced Computer Science and Applications, 11(4), https://doi.org/10.14569/IJACSA.2020.0110403

[166] Adadi, A., et al (2015). Semantic web service composition for e-government services. Journal of Theoretical and Applied Information Technology, 71(3), http://www.jatit.org/volumes/Vol71No3/2Vol71No3.pdf

[167] Shah, R. T., & Patel, S. V. (2016). Survey on issues and challenges of web service development, composition, discovery. VNSGU Journal of Science and Technology, 5(1), 146-158

[168] Charfi, A., et al (2009). A plug-in architecture for self-adaptive web service compositions. In 2009 IEEE International Conference on Web Services (pp. 35-42). IEEE. https://doi.org/10.1109/ICWS.2009.19

[169] Razian, M., et al (2022). Service composition in dynamic environments: A systematic review and future directions. Journal of Systems and Software, 188, 111290. https://doi.org/10.1016/j.jss.2022.111290

[170] Acampora, G., et al (2010). Interoperable and adaptive fuzzy services for ambient intelligence applications. ACM Transactions on Autonomous and Adaptive Systems, 5(2), Article 8. https://doi.org/10.1145/1740600.1740604

[171] Brønsted, J., et al (2010). Service composition issues in pervasive computing. IEEE Pervasive Computing, 9(1), 62-70. https://doi.org/10.1109/MPRV.2010.11

[172] Adala, A., et al (2011). A framework for automatic web service discovery based on semantics and NLP techniques. Advances in Multimedia, 2011(1)

[173] Hamzei, M., et al (2023). A quality-of-service-aware service composition method in the Internet of Things using a multi-objective fuzzy-based hybrid algorithm. Sensors, 23(16), 7233. https://doi.org/10.3390/s23167233

[174] Charlet, J., et al (2004). Ontologies pour le Web sémantique [Ontologies for the semantic web]. Revue Information, Interaction, Intelligence (I3), 4(2)

[175] Rao, V., et al (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303–315.

[176] eGOV Project. (n.d.). eGOV project website. http://www.egovproject.org

[177] Kavadias, G., & Tambouris, E. (2003). GovML: A markup language for describing public services and life events. In Working Conference on Knowledge Management in Electronic Government. Springer.

[178] Moumtzi, V., et al (2009). Collaborative e-gov networks: The case of the Semantic-Gov project. In Integrating e-business models for government solutions: Citizen-centric service oriented methodologies and processes. IGI Global. https://-doi.org/10.4018/978-1-60566-240-4

[179] TERREGOV Consortium. (2005). Definition of TERREGOV prototype (Deliverables 4.2 & 4.3). http://terregov.eupm.net/

[180] Vasiliu, L., & Harand, S. (2005). The DIP project: Putting semantics into web services. In 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT 2005)

[181] OntoGov Project. (n.d.). OntoGov project website. http://www.ontogov.com

[182] Gabel, T., et al (2004). KAON - ontology management infrastructure (SEKT Deliverable D3.1.1.a). SEKT Project.

[183] Hobold, G. C., & Siqueira, F. (2012). Discovery of semantic web services compositions based on SAWSDL annotations. In 2012 IEEE 19th International Conference on Web Services (pp. 280-287). IEEE. https://doi.org/10.1109/ICWS.2012.24

[184] Wang, H., et al (2009). Query for semantic web services using SPARQL-DL. In 2009, Second International Symposium on Knowledge Acquisition and Modeling (pp. 367–370). IEEE. https://doi.org/10.1109/KAM.2009.198

[185] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199-220. https://doi.org/10.1006/knac.1993.1008

[186] Bettahar, F., et al (2009). Towards a semantic interoperability in an e-government application. Electronic Journal of e-Government, 7(3), 209-226.

[187] Gómez-Pérez, A. (1996). Towards a framework to verify knowledge sharing technology. Expert Systems with Applications, 11(4), 519-529. https://-doi.org/10.1016/S0957-4174(96)00068-5

[188] Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology (Technical Report KSL-01-05). Stanford Knowledge Systems Laboratory.

[189] Bravo, M., et al (2019). Methodology for ontology design and construction. Contaduría y Administración, 64(4), 1-24. https://-doi.org/10.22201/fca.24488410e.2019.1987

[190] Sattar, A., et al (2020). Comparative analysis of methodologies for domain ontology development: A systematic review. International Journal of Advanced Computer Science and Applications, 11(5), 99-108. https://doi.org/10.14569/I-JACSA.2020.0110512

[191] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199-220. https://doi.org/10.1006/knac.1993.1008

[192] Sathya, M., et al (2010). Evaluation of QoS-based web-service selection techniques for service composition. International Journal of Software Engineering, 1(5), 73-90.

[193] Merzoug, et al(2014). QoS-aware web service selection based on harmony search [Conference presentation]. International Symposium on ISKO-Maghreb, Algiers, Algeria.

[194] Alrifai, M., et al (2011). A hybrid approach for efficient web service composition with end-to-end QoS constraints. ACM Transactions on the Web, 5(2), Article 7. https://doi.org/10.1145/1961659.1961664

[195] Yu, Q., & Bouguettaya, A. (2010). Foundations for efficient web service selection. Springer.

[196] Hadjila, F. (2011). QoS-aware service selection based on genetic algorithm. In Proceedings of CIIA'11 (pp. [page range]). Saida, Algeria.

[197] Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning (pp. 760-766). Springer. https://doi.org/10.1007/978-0-387-30164-8_630

[198] Mousa, A., & Bentahar, J. (2016). An efficient QoS-aware web services selection using social spider algorithm. Procedia Computer Science, 94, 176-184. https://doi.org/10.1016/j.procs.2016.08.028

[199] Alsamara, K., et al. (2022). Administrative organization of healthcare institutions in Algeria: Between centralization and decentralization. Open Access Macedonian Journal of Medical Sciences, 10(E), 1114-1118. https://doi.org/10.3889/oamjms.2022.9509

[200] Rezki, K., & Fateh, D. (2024). The reality of health development in Algeria in light of the current challenges. Journal of Legal and Economic Research, 7(1), 408-431.

[201] Abbou, Y., & Brahamia, B. (2017). Le système de santé algérien entre gratuité des soins et maîtrise des dépenses de santé [The Algerian health system between free care and health expenditure control]. Insaniyat, 75-76, 149-171.

[202] Kernane, A., & Mordjane, I. (2024). Digital revolution in Algerian public health: Towards optimized management of resources and care. Journal of Social Protection Research, 5(2), 52-74.

[203] Boufissiou, S. (2024). Analytical sight on the Algerian digital health strategy 2023 - 2027. Akofena, 14(2)

[204] Djouab, H. (2024). Rôle of digitalization in improving health services in hospitals in Algeria. European Economic Letters, 14(3), [page range]. ISSN 2323-5233.

[205] Jardim, S. V. B. (2013). The electronic health record and its contribution to healthcare information systems interoperability. Procedia Technology, 9, 940-948. https://doi.org/10.1016/j.protcy.2013.12.105

[206] International Telecommunication Union. (2012). E-health standards and interoperability (ITU-T Technology Watch Report). https://www.itu.int/en/ITU-T/techwatch/Pages/ehealth-standards.aspx

[207] Ait Abdelouahid, R., et al (2023). Literature review: Clinical data interoperability models. Information, 14(7), 364. https://doi.org/10.3390/info14070364

[208] Introduction to Semantic Interoperability, KNOWLEDGE TOOLS, Pan American Health Organization, 2021, https://iris.paho.org/handle/10665.2/55417, Accessed on: 10 September 2024

[209] Healthcare Information and Management Systems Society. (n.d.). What is interoperability? https://www.himss.org/resources/interoperability-healthcare

[210] Lee, A. R., et al (2021). Developing a transnational health record framework with level-specific interoperability guidelines based on a related literature review. Healthcare, 9(1), 67. https://doi.org/10.3390/healthcare9010067

[211] Adel, E., et al (2021). An extended semantic interoperability model for distributed electronic health record based on fuzzy ontology semantics. Electronics, 10(14), 1733. https://doi.org/10.3390/electronics10141733

[212] Adel, E., et al (2022). Semantic architecture for interoperability in distributed healthcare systems. IEEE Access, 10

[213] Bhartiya, S., & Mehrotra, D. (2014). Challenges and recommendations to healthcare data exchange in an interoperable environment. Electronic Journal of Health Informatics, 8(1), 16.

[214] Gansel, X., et al (2019). Semantic data interoperability, digital medicine, and e-health in infectious disease management: A review. European Journal of Clinical Microbiology & Infectious Diseases, 38(6), 1023-1034. https://-doi.org/10.1007/s10096-019-03539-6

[215] Salomi, M. J. A., & Claro, P. B. (2020). Adopting healthcare information exchange among organizations, regions, and hospital systems toward quality, sustainability, and effectiveness. Technology and Investment, 11(3), https://www.scirp.org/html/3-9900884_101997.htm

[216] do Espírito Santo, J. M., & Medeiros, C. B. (2017). Semantic interoperability of clinical data. In M. Da Silveira, C. Pruski, & R. Schneider (Eds.), Data integration in the life sciences (Vol. 10649, pp. 29-37). Springer. https://doi.org/10.1007/978-3-319-69751-4_3

[217] Bernaldo de Quirós, E. G., et al. (2018). Terminology services: Standard terminologies to control health vocabulary. IMIA Yearbook of Medical Informatics, 2018(1), 227-233. http://dx.doi.org/10.1055/s-0038-1641200

[218] Liyanage, H., et al (2015). Using ontologies to improve semantic interoperability in health data. Journal of Innovation in Health Informatics, 22(2), 309-315. https://doi.org/10.14236/jhi.v22i2.159

[219] Haendel, M. A., et al. (2018). Classification, ontology, and precision medicine. New England Journal of Medicine, 379(15), 1452-1462. https://doi.org/10.1056/NEJMra1615014

[220] El Hajjamy, O., et al. (2018). Integration of heterogeneous classical data sources in an ontological database. In International conference on big data, cloud and applications (pp. 417-432). Springer.

[221] Calcagno, S., et al. (2024). Merging ontologies and data from electronic health records. Future Internet, 16(2), 62. https://doi.org/10.3390/fi16020062

[222] Kiourtis, A., et al (2017, June). Aggregating heterogeneous health data through an ontological common health language. In 2017 10th International Conference on Developments in eSystems Engineering (DeSE) (pp. 175-181). IEEE.

[223] Kone/Tapsoba, L. S., et al. (2023). Interoperability approach for hospital information systems based on the composition of web services. Procedia Computer Science, 219, 1161-1168.

[224] Urov, V., et al. (2012). An agent coordination framework for IHE based cross-community health record exchange. In A. Moreno & D. Isern (Eds.), International conference on autonomous agents and multiagent systems (AAMAS) (pp. 29-39). IFAAMAS.