Knowledge-based Life Event Model for E-government Service Integration with Illustrative Examples

Abdullah Alqahtani, Haiyan Lu and Jie Lu
Decision System & e-service Intelligence (DeSl) Lab
Center for Quantum Computation and Intelligent Systems (QCIS)
Faculty of Engineering and Information Technology
University of Technology Sydney, Australia
Abdullah.AlQahtani@student.uts.edu.au, Haiyan.Lu, Jie.Lu@uts.edu.au

Abstract. The advancement of information and communications technology and web services offers an opportunity for e-government service integration, which can help improve the availability and quality of services offered. However, few of the potential service integration applications have been adopted by governments to increase the accessibility of and satisfaction with government services and information for citizens. Recently, the 'life event' concept was introduced as the core element of integrating complexity of service delivery to improve the efficiency and reusability of e-government services, web-based information management systems. In addition, a semantic web-based ontology is considered to be the most powerful conceptual approach for dealing with challenges associated with developing seamless systems in distributed environments. Among these challenges are interoperability, which can be loosely defined as the technical capability for interoperation. Despite the conceptual emergence of semantic web-based ontology for life events, the question remains of what methodology to use when designing a semantic web-based ontology for life events. This paper proposes a semantic web-based ontology model for life events for e-government service integration created using a methodology that implements the model using the ontology modelling tool Protégé and evaluates the model using Pellet Reasoner and the SPARQL query language. In addition, this model is illustrated by two examples, the Saudi Arabia King Abdullah Scholarship and Hafiz, to show the advantages of integrated systems compared with standalone systems. These examples show that the new model can effectively support the integration of standalone e-government services automatically so that citizens do not need to manually execute individual services. This can significantly improve the accessibility of e-government services and citizen's satisfaction.

Keywords*:* Ontology, e-government service integration, life event.

1. INTRODUCTION

E-government services make use of ICT (Information and Communication Technology), the World Wide Web or the internet to deliver information and services to citizens and other agencies [1, 26]. Rapid development of ICT and the advancement of semantic web service techniques offer great potential for improving the efficiency of web services [27]. On the other hand, e-government service integration improves the availability and quality of services offered. The vision of e-government service integration is to automatically discover, select, composite and execute suitable service components across various government domains to serve the citizens' needs. Semantic web service technologies offer an opportunity for e-government service integration that can improve the availability, reusability and quality of services offered [15]. However, few of the potential applications of e-government service integration have been adopted and implemented by semantic web-based ontology frameworks such as OWL-S (Ontology Web Language for Services) framework and WSMO (Web Service Modelling Ontology) [13, 17, 18]. Those applications adopted OWL-S or WSMO frameworks as a service ontology for modelling public services, but those frameworks don't enable the complete reusability of services. Moreover, most e-government service systems offer standalone rather than integrated services [3]. This has resulted in redundant ICT systems, great inconvenience for users and the low efficiency of e-government service delivery. This not only undermines the rationale for using ICT and e-government but also creates heavy overheads in e-government.

To overcome these problems, a popular development trend in e-government service integration is the use of life events, which is a set of actions with at least one action in the public service domain. When this set of actions is executed in the appropriate workflow, citizens' needs arising from new life situations are met [19, 33]. Several solutions have been proposed based on life events [4, 7, 22, 25 29, 30, 31, 33]. Analysis of these solutions shows that their focus is on defining the life event's approaches, technical level and regulations.

Despite the successful use of semantic web-based ontology for life events to offer seamless integration in various e-government systems, the question remains of what methodology to use to design semantic web-based ontologies for life events. In addition, the presentation of life event concepts and service concepts separately make life-event-based e-government service integration systems seem quite complicated and abstruse.

This paper proposes a new effective model for e-government service integration based on a semantic web-based ontology to address the above issues. More specifically, this model used a semantic web-based ontology to improve workflow in life-event-based e-government service integration and to enable a complete reusability of services using a life event ontology (LEO) as a database to store and manage workflow. This paper uses an ontology building methodology to facilitate designing a semantic web-based ontology of life events for e-government service integration, implementing the model using the ontology modelling tool Protégé and evaluating the model using Pellet Reasoner and the SPARQL query language. In addition, this model is illustrated by two examples, the Saudi Arabia King Abdullah Scholarship and Hafiz, to show the advantages of integrated systems compared with standalone systems. These examples show that new models can effectively support the integration of standalone e-government services automatically, so that citizens do not need to manually execute individual services. The study also aims to improve the accessibility of e-government services and citizen's satisfaction.

The rest of this paper is organized as follows: Section 2 presents the related work; Section 3 presents the literature review; Section 4 presents the development, implementation and evaluation of LEO-based e-government service integration using an ontology building methodology; Section 5 presents illustrative examples; and Section 6 concludes the paper and recommends future work.

2. Literature Review

Knowledge management systems are managerial systems and approaches used for the collection, processing and organization of knowledge assets. In an environment where data is from distributed sources, metadata is required for the annotation and modelling of relations between the information objects [21]. Choosing an appropriate conceptual representation, which serves as the basis for collection of metadata and processing, is an important step to take before the actual development of such systems. E-government systems can be looked at as knowledge systems, as they allow for the collection, processing and organization of information and knowledge from citizens and government departments. There are two major approaches that can be used to conceptually represent the systems and aid the collection of metadata. The first approach uses a taxonomy, or a hierarchical model of classes and subclasses, that represents ideas and terms of certain conceptual scope. The other approach is the use of a thesaurus, which is like a taxonomy, that provides hierarchical relations and systematically ordered lists of terms that form a clearly defined domain [21]. It is worth noting that both approaches are conceptually neutral with respect to further definition and description of their content. In both cases, there is no effort to further describe or define the content.

Ontology is a formal representation of knowledge as sets of concepts within domains and the relationships between the represented concepts [8]. Ontology can therefore be used to describe entities within a domain. An ontology as a shared conceptualization renders a shared vocabulary and taxonomy that can be used to model domains with definitions of objects, their relationships and properties. Thus, ontology can be looked at as a structural framework that allows for the organization of information. Ontologies are considered to be the most powerful conceptual approach for dealing with challenges associated with developing seamless systems in distributed environments [2]. Ontologies also provide a way of representing formal and shared domain descriptions that allow for conceptualization of challenges and solutions in modelling within distributed environments [24]. Despite the existence of both formal and informal definitions of the term ontology, there is a general convergence on the importance of a shared domain knowledge, which is the main aspect in the use of ontologies.

Research studies reveal that ontologies have been extensively used in efforts to improve the use of e-government in public service [15]. Some of the ontologies that have been used extensively include the idea of a semantic web. For example, the eGOV project is an attempt to develop an architecture that would enable the description of services as part of a one-stop government [35]. The project involved the development of a mark-up language (GovML) that defined a set of metadata to describe services offered to the public and life events [14]. Efforts aimed at the automation of administrative processes (in cases of multiple administrations) and supporting reuse of data led to the eGovSM project [20]. The project developed a set of XML schema models that have the capability to support interoperable systems. However, neither project takes into account the use of the semantic web for the interoperability and integration of different web applications in the e-government domain. Further, these projects mainly focus on the technical level and do not provide detailed guidelines for developing the e-government service integration.

The semantic web-services-based ontology mission is to combine web service (WSDL, SOAP) technologies with semantic web technologies (RDF, RDFS, OWL), to support automatic and dynamic interaction between various software systems [27]. Although web services technology is the standard way of describing interfaces, it does not explain anything about what software systems do or what sequences of messages are used in the process of interaction. By using semantic web technology, we overcome this obstacle. The use of semantic web-to-web services has led to the rise of semantic web services [5]. The rise of this new technology has made it easier for software systems to define, find and execute web services automatically without instructions from people [28]. This technology offers an opportunity for e-government service integration that can help improve the availability, reusability and quality of services offered. Although semantic web-based ontology frameworks such as the OWL-S (Ontology Web Language for Services) framework have been used to potential applications of e-government service integration as shown in [13, 17, 18], those frameworks don't enable the complete reusability of services.

A life event is defined as a set of actions, with at least one in the public service domain. When this set of actions is executed in the appropriate workflow, citizen needs that arise from new life situations are met [19, 33]. The use of single entry points to integrated services from different public institutions improves the usability and potential gains from the e-government systems. The negative effects caused by users' lack of knowledge on how to meet their needs when interacting with e-government systems can be minimized if the services are based on life events.

A critical analysis of existing literatures show that there are two main approaches used in modelling life events. Under the first approach, life events are modelled as workflows of related actions and services based on citizen profiles [30]. The second approach involves the use of ontologies in modelling life events [22]. Under the second approach, the notion of semantic representation of knowledge is extensively utilised [22]. A number of researchers have used the second approach in their respective fields. For example, Trochidis, Tambouris and Tarabanis [31] suggested a new model for an LEO and used that model in onestopgov. In addition, Sanati and Lu [25] suggested a multilevel modelling framework for analysis and design of life events in government services integration while focusing on a regulation perspective. Although the important life event in e-government service integration, current research focuses on defining life events' approaches, defining models of life events' levels and features and a conceptual model of LEO-based e-government service integration [25, 29, 30, 31, 33]. These studies indicate that most research of life-event-based e-government service integration is still lacking in detailed guidelines for developing LEO-based e-government service integration. Therefore, this paper suggests the development of an LEO for e-government services integration based on semantic ontology and an ontology-building methodology. In addition, this paper uses a semantic web-based ontology to improve workflow in life-event-based e-government service integration through the use of an LEO as a database to store and mange workflow.

3. Saudi Arabia as Case Study

This paper addresses Saudi Arabia as a case study to verify the proposed life event model. To understand our case study, we offer two examples of life events: the Saudi Arabia King Abdullah scholarship program (SAKASP) and the Hafiz Program. SAKASP is a good case study for failed integration of e-government services. The program offers financial support for eligible Saudi Arabian post-graduate students. Even though online submission is possible, the collection and verification of the required information is manual [1]. The two main players involved in the program are the student applicants and the commission of the ministry of higher education. In general, the process of applying for a scholarship has two shortcomings that are typical of non-integrated e-government systems. First, applicants have to communicate with and visit three ministries when applying for the scholarship. Second, the commission manually processes the application. These requirements result in delays and inconvenience for the applicants and renders the entire application process inefficient. The choice of the program as a case study is thus guided by the fact that it involves various ministries (e-government systems) and displays problems that are characteristic of non-integrated or poorly integrated e-government systems. The second example is the Hafiz Program, which is a national program to help job seekers. According to the SAUDI National e-Government Portal, the Hafiz program gives financial aid to job seekers who conform to the program's criteria [11]. The Ministry of Labour is responsible for completing the reception and registration of applications to support researchers working in Saudi Arabia. In this program, there is also a service integration problem throughout different ministries. In this paper, we considered these case studies as examples to verify our life event model.

4. DEVELOPMENT AND IMPLEMENTATION OF LIFE EVENT ONTOLOGY FOR INTEGRATING MULTIPLE WEB SERVICES FROM DIFFERENT WEB APPLICATIONS

The life event as an ontology was developed using Methontology to enable the complete reusability of services. Methontology is a structured method for building ontologies that is based on experience acquired in developing ontologies [28]. A complete explanation of Methontology is beyond the scope of this paper, but further information on how to use Methontology to develop ontologies can be found in Corcho et al. [6] and Fernández-López, Gómez-Pérez and Juristo [9].

As a result, the concepts and verbs have been optimized and are highly relevant to the specific case study. Concepts from the semantic web were integrated into the ontology. This ensures that the life event model used maximizes the strengths associated with different ontologies and minimizes individual weaknesses. The use of Methontology in developing the life event model specific to the case study is a strength that allows for the formulation of an ontology that harnesses the benefits and strengths of other ontologies. An ontology development language and ontology development tool for LEO construction was also selected for two reasons. First, several one-to-many relations exist in the proposed LEO. Therefore, the full expressiveness of cardinality restrictions requires the use of OWL DL instead of OWL Lite. The second reason is the existence of OWL DL reasoners. OWL DL has maximum expressiveness while retaining computational completeness. In addition, we chose Protégé-OWL as a development tool for designing an LEO because it is supported by Methontology and is widely used because of its platform-independent characteristics.

LEO-building processes contain specific activities: specification, conceptualization, formalization, implementation and evaluation.

4.1 Building Process: Specification

To achieve specification activity, we carried out several sub-activities, such as domain analysis, consideration of existing systems and knowledge acquisition. The target domain of our LEO is for integrating multiple e-government services from different web applications to achieve one life event by citizen. The LEO needs to capture all the information relevant to life event processes, such as applying for scholarships. The life event process is so complex because various application requirements need to be checked. These life event process requirements include the list of required services, rules to be taken, the number of inputs from users and so on. The ultimate purpose of the LEO is to determine whether the applicant can apply and to inform the user what requirements need to be fulfilled according to their life event process requirements.

During the domain analysis, one way to support the determining purpose of the ontology and the scope of the ontology is to sketch a list of questions, ideally based on competency, that a knowledge base derived from the ontology should be able to answer [12]. The competency questions should be related to the specified purpose of the ontology and scope of the ontology. To define the LEO's purpose and scope, the researchers started by asking themselves competency questions (e.g., What domain will the ontology cover? What types of questions should the LEO answer? Who will be the main users of the LEO?), as suggested by Uschold and Gruninger [32]. The purpose of the LEO is to facilitate access to e-government services. We believe the questions it should be able to answer are:

A. What are the services provided by the government?
B. Which government agency (ministry) provides each service?
C. What are the general life events provided by the government?
D. What are event steps that can handle life event's workflow?
E. Which government agency (ministry) provides each life event?
F. For each event step, what life event might naturally call for to its utilization? or equivalently: for each life event, what event steps are relevant?
G. For each service, what event step might naturally call for to its utilization? (or equivalently: For each service, what event step is relevant?)
H. What are the inputs required to execute the service?
I. What outputs come out from the service?
J. What rules will be implemented on output class?

Judging by this list of questions, the ontology will include information about life events, all services that are related to specific life events, the citizen who faces the life event, the input and output that is used by each service and the rules that are applied on the input and output for each service.

Target users for the LEO are citizens, ministries and administration staff responsible for the life event process. They should be able to identify all personal information (such as educational history) of a citizen whenever they want to. Furthermore, they should be able to detect what other qualifications should be fulfilled to meet the life event requirements (such as requirements for a scholarship application).

4.2 Building Process: Conceptualization.

Based Based on the relevant literature and analysis system, and in order to extend the value of the LEO, a number of concepts were identified for describing an LEO for integrating multiple e-government services from different web applications. These concepts are life event, event step, service, input, output, provider and rule.

i. Life Event

The life event class is designed to answer the questions 'For each life event, what services are relevant?' and 'What are the general life events provided by the government?' The Life Event class includes all the citizen's wishes and needs, derived from multiple events and aspects of his/her life situation, such as apply SAKASP.

ii. Event Step

The event step class is designed to answer the questions 'What are event steps that can handle life event's workflow?' and 'For each life event, what event steps are relevant?' The Event Step class collects sets of event steps that must be performed to resolve the particular life event and organises life events as event steps.

iii. Service

The service class is designed to answer the questions 'For each service, what event step might naturally call for to its utilization?' and 'What are the services provided by the government?' The Service class collects sets of services that must be performed to resolve each particular event step. It presents atomic activities provided by Ministries.

iv. Input

The input class is designed to answer the question 'What inputs are required to use a given service?' This class specifies the input parameters used by services.

v. Output

The output class answers the question 'What outputs come from the service?' This class specifies the output parameters that present the decision of the service provider regarding the service (web service).

vi. Provider

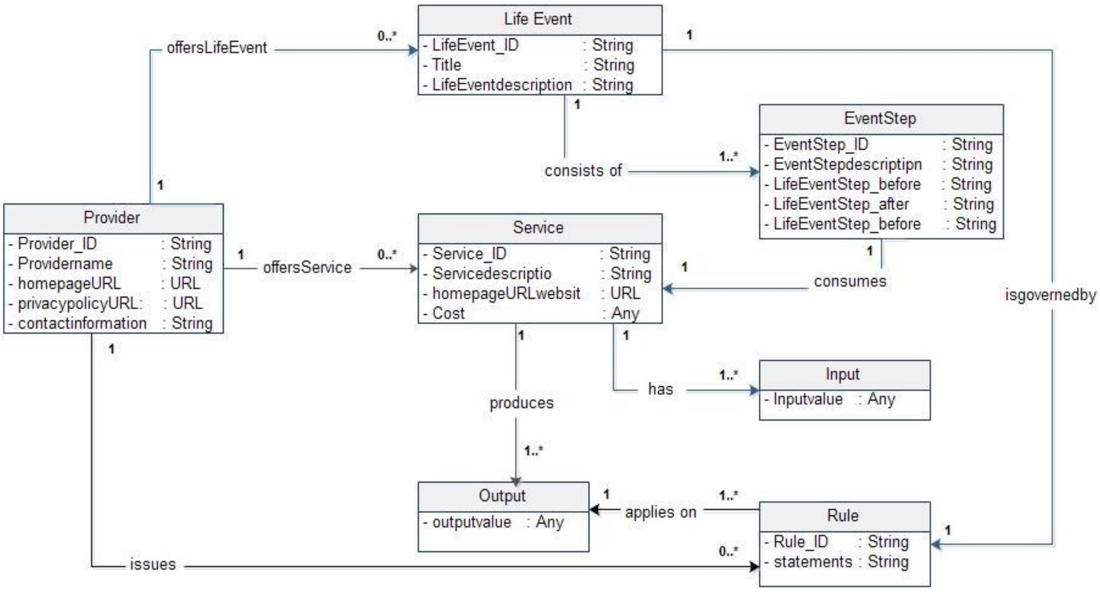
The provider class is designed to answer the questions 'Which government agency (ministry) provides each service?' and 'Which government agency (ministry) provides each life event?' This class defines which governmental division is responsible for defining and/or offering the life event and service to the citizen.

vii. Rule

The rule class is designed to answer the question 'What are rules implemented on output class?' In this class, there are rules that issue from the life event's provider and implements on output class.

Based on the above concepts, the relationships between the entities in our LEO are shown in Fig. 1 to define the relationships between the source and target concepts.

Fig. 1: THE RELATIONSHIP BETWEEN THE ENTITIES IN THE LIFE EVENT ONTOLOGY



The next task was to create concepts and their properties table in detail to include concepts, object properties and datatype properties, as shown in Table 1.

In this phase, we defined instances of concepts existing in the e-government service and presented examples of instances, as shown in Table 2.

TABLE 1: CONCEPTS AND THEIR PROPERTIES TABLE

Concepts	**Object properties**	**Datatype properties**
Life event	Consistof LifeEventisgovernedby LifeEventisprovidedby	hasLifeEvent_ID hastitle hasLifeEventdescription
Input		hasInputvalue
Output	Outputisgovernedby	hasOutputvalue
Event Step	consumes	hasEventSteporder_ID hasEventStepdescription hasEventStep_before hasEventStep_after hasEventStep_status
Provider	offersService issuesRule offersLifeEvent	hasProvider_ID hasProvidername hascontactinformation hashomepageURL hasprivacypolicyURL
Service	hasInput producesOutput isprovidedby	hasService_ID hasServicedescription hasWSDLURLwebsite hascost
Rule	appliesonOutput	hasRule_ID hasstatements

TABLE 2: INSTANCES OF CONCEPTS

Class	Examples about Instance
Life event	Apply SAKASP_1 - Apply hafiz_2
Input	Check Commercial Activity Service Citizen ID – Check Commercial Activity Service Citizen Password – Check Job Status Service Citizen ID
Output	Check Job Status Service Result – Check Mark Service Result - Check Personal Information for Hafiz Service Date of Birth Result - Check an Employee in the Private Sector Service Citizen Job Record Result
EventStep	heck Job Status step_1 - Check Mark step_2 - Check Personal Information for Scholarships step_3 - Check Job Status step_4 - Check Personal Information for Hafiz Step_5
Provider	Ministry of Higher Education_1 - Ministry of Labour_2 - Ministry Of Civil Service_1 - National Centre For Assessment in Higher Education_2 - Ministry of Civil Affairs_3
Service	Check Job Status Service_1 - Check Mark Service_2 - Check Personal Information for Scholarships Service_3 - Check Personal Information for Hafiz Service_4
Rule	Apply SAKASP_Check Job Status Service Rule - Apply SAKASP_Check Mark Service Rule - Apply hafiz_Check Job Status Service Rule

4.3 ***Building Process:*** *Formalization and Implementation.*

Our conceptualized LEO was formalized and implemented. We used Protégé-OWL to convert our formal model into an OWL-DL. Protégé-OWL supports graphical representation of a class hierarchy though the OWLVIZ plug-in. The LEO consists of seven classes: life event, input, output, provider, service, rule and event step. Fig. 2 shows the hierarchy of the LEO through the OWLVIZ plug-in. However, this plugin provides a class-level view only and does not provide views for relationships (object properties). Therefore, we used the Jambalaya Plugin to provide views for relationships between concepts in our LEO, as shown in Fig. 3.

Fig. 2: CLASSES

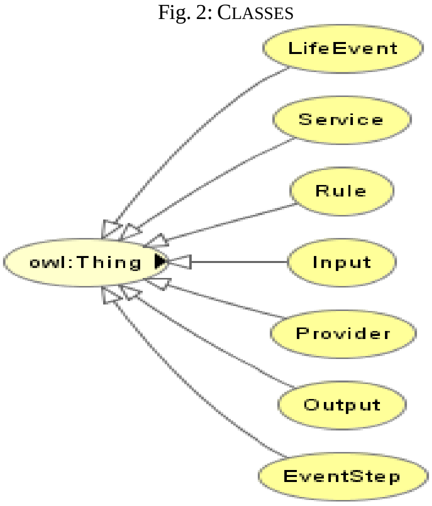
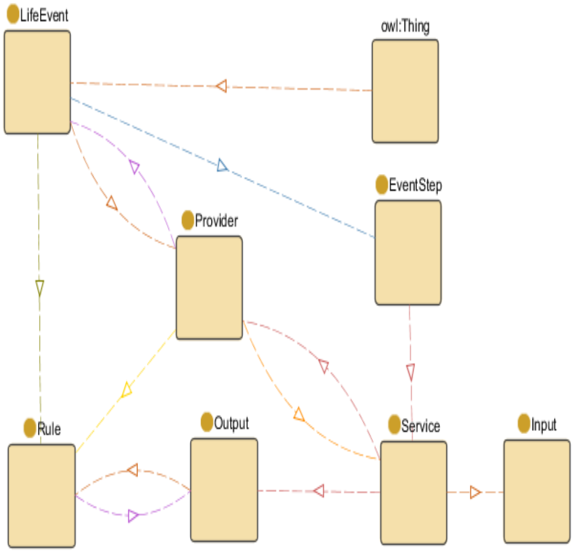
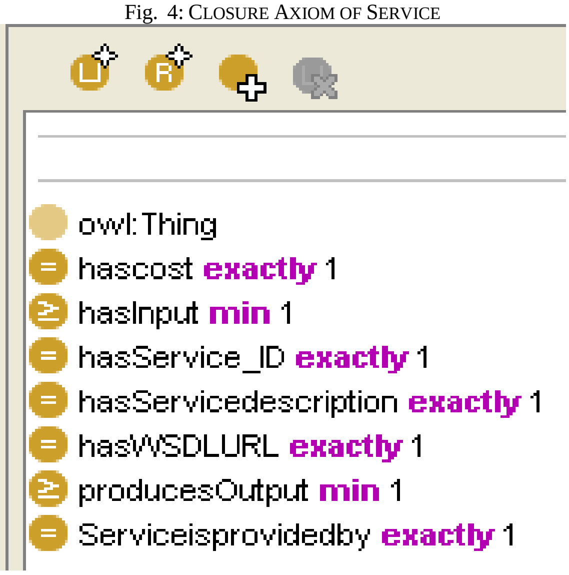


Fig. 3: THE RELATIONSHIPS BETWEEN CLASSES



The notions behind our LEO have been described above and we have defined a concept of an LEO with a number of object properties and datatype properties. We used the 'closure axiom' notion to express the definition and the axiom of a specific class as shown in Fig. 4 as an example.

Fig. 4: CLOSURE AXIOM OF SERVICE


owl:Thing
hascost exactly 1
hasInput min 1
hasService_ID exactly 1
hasServicedescription exactly 1
hasWSDLURL exactly 1
producesOutput min 1
Serviceisprovidedby exactly 1

4.4 ***Building Process:*** *Evaluation*

Ontologies should be evaluated before they are used or reused. According to Gómez-Pérez, Fernández-López and Corcho [10] ontology evaluation is composed of two steps: ontology verification and ontology validation.

The first step is ontology verification, or ensuring that the ontology has been built correctly according to certain specified ontology quality criteria. The criteria used in the verification step are consistency, completeness and conciseness [10]. For the verification step, we used two methods. The first method uses verification tools for OWL modelling, such as those from the University of Manchester. We used the Manchester OWL validator to verify the LEO. According to the output from the Manchester OWL validator, the LEO conforms to the OWL2 DL profile. The second method uses the Pellet Reasoner inside Protégé. When we ran the Pellet Reasoner inside Protégé, we found some 'Bad' individuals.

The second step is validation through the SPARQL query, which answers competency questions from the ontology development model [15, 23, 34]. To evaluate the competency using the SPARQL query, we formalized our competency question as a SPARQL query. Then, we compared the actual and the expected results. In Table 3, we show examples of questions and results for different set queries in the SPARQL query.

TABLE 3: THE VALIDATION TASK

Competence Questions	SPARQL query SPARQL	query's answer

A. What services are provided by the government?	PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl#>
	SELECT ?service
	WHERE { ?service rdf:type :Service }
		Fig. 5

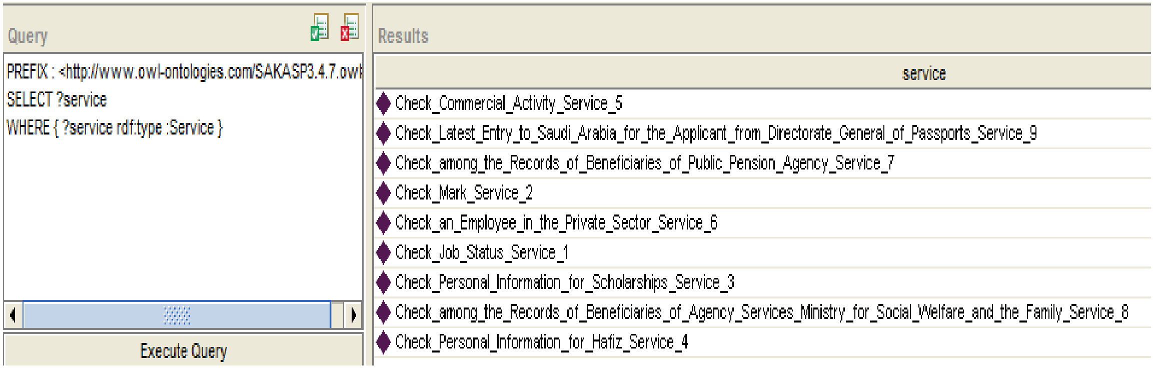
B. What event steps can the handle life event's workflow?	PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl#>
	SELECT ?EventStep
	WHERE { ?EventStep rdf:type :EventStep }
		Fig. 6

C. For each event step, what life event might naturally call for its utilization? Or, for each life event, what event steps are relevant?	PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl#>
	SELECT ?LifeEvent ?EventStep
	WHERE { ?LifeEvent :consistsofEventStep ?EventStep }
		Fig. 7

D. For each service, what event step might naturally call for its utilization? Or, for each service, what event step is relevant?)	PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl#>
	SELECT ?Service ?EventStep
	WHERE { ?EventStep :consumes ?Service }
		Fig. 8

At Fig. 5-8 illustrate questions A, B, C and D formulated as queries in SPARQL query.

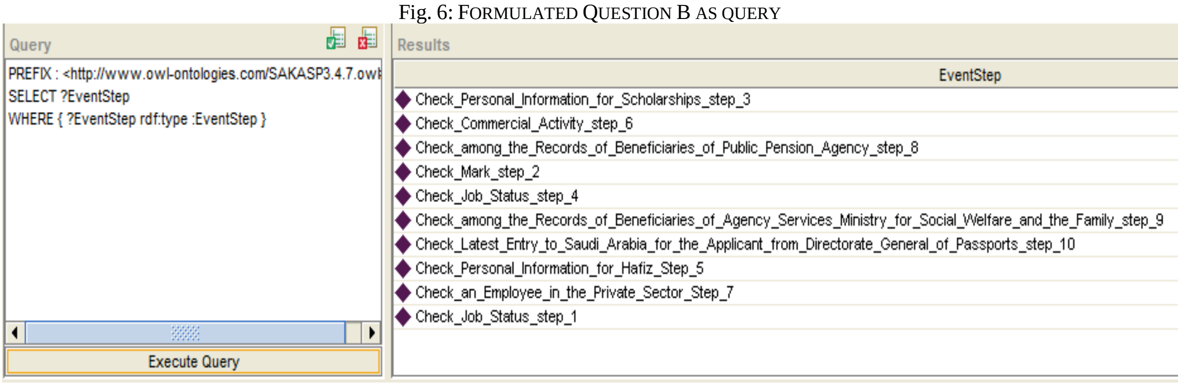
Fig. 5: FORMULATED QUESTION A AS QUERY



```
PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl
SELECT ?service
WHERE { ?service rdf:type :Service }
```

service
Check_Commercial_Activity_Service_5
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_9
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_7
Check_Mark_Service_2
Check_an_Employee_in_the_Private_Sector_Service_6
Check_Job_Status_Service_1
Check_Personal_Information_for_Scholarships_Service_3
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_Service_8
Check_Personal_Information_for_Hafiz_Service_4

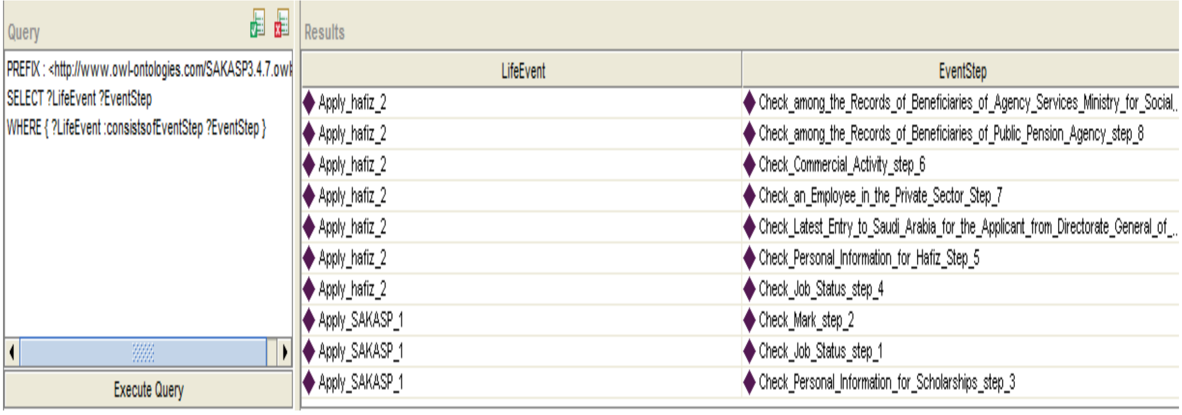
Fig. 6: FORMULATED QUESTION B AS QUERY



```
PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl
SELECT ?EventStep
WHERE { ?EventStep rdf:type :EventStep }
```

EventStep
Check_Personal_Information_for_Scholarships_step_3
Check_Commercial_Activity_step_6
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_step_8
Check_Mark_step_2
Check_Job_Status_step_4
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_step_9
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_step_10
Check_Personal_Information_for_Hafiz_Step_5
Check_an_Employee_in_the_Private_Sector_Step_7
Check_Job_Status_step_1

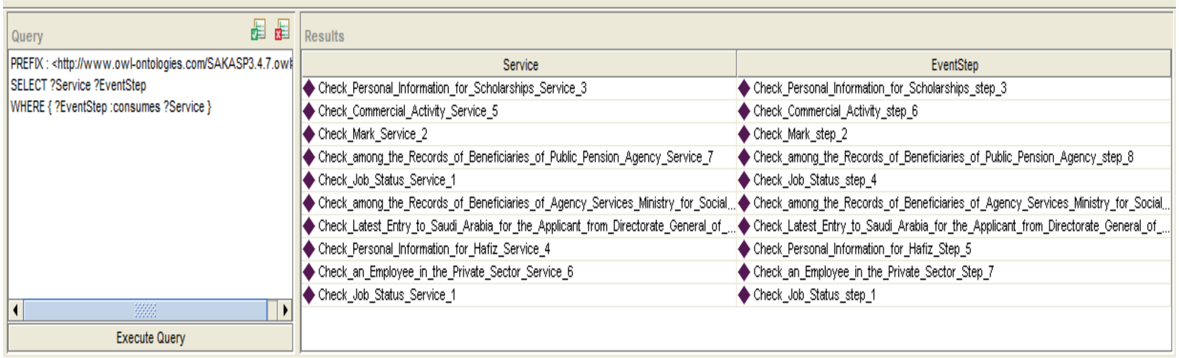
Fig. 7: FORMULATED QUESTION D AS QUERY



```
PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl
SELECT ?LifeEvent ?EventStep
WHERE { ?LifeEvent :consistsofEventStep ?EventStep }
```

LifeEvent	EventStep
◆ Apply_hafiz_2	◆ Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_
◆ Apply_hafiz_2	◆ Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_step_8
◆ Apply_hafiz_2	◆ Check_Commercial_Activity_step_6
◆ Apply_hafiz_2	◆ Check_an_Employee_in_the_Private_Sector_Step_7
◆ Apply_hafiz_2	◆ Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_..
◆ Apply_hafiz_2	◆ Check_Personal_Information_for_Hafiz_Step_5
◆ Apply_hafiz_2	◆ Check_Job_Status_step_4
◆ Apply_SAKASP_1	◆ Check_Mark_step_2
◆ Apply_SAKASP_1	◆ Check_Job_Status_step_1
◆ Apply_SAKASP_1	◆ Check_Personal_Information_for_Scholarships_step_3

Fig. 8: FORMULATED QUESTION C AS QUERY



```
PREFIX : <http://www.owl-ontologies.com/SAKASP3.4.7.owl
SELECT ?Service ?EventStep
WHERE { ?EventStep :consumes ?Service }
```

Service	EventStep
◆ Check_Personal_Information_for_Scholarships_Service_3	◆ Check_Personal_Information_for_Scholarships_step_3
◆ Check_Commercial_Activity_Service_5	◆ Check_Commercial_Activity_step_6
◆ Check_Mark_Service_2	◆ Check_Mark_step_2
◆ Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_7	◆ Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_step_8
◆ Check_Job_Status_Service_1	◆ Check_Job_Status_step_4
◆ Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social...	◆ Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social...
◆ Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_...	◆ Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_...
◆ Check_Personal_Information_for_Hafiz_Service_4	◆ Check_Personal_Information_for_Hafiz_Step_5
◆ Check_an_Employee_in_the_Private_Sector_Service_6	◆ Check_an_Employee_in_the_Private_Sector_Step_7
◆ Check_Job_Status_Service_1	◆ Check_Job_Status_step_1

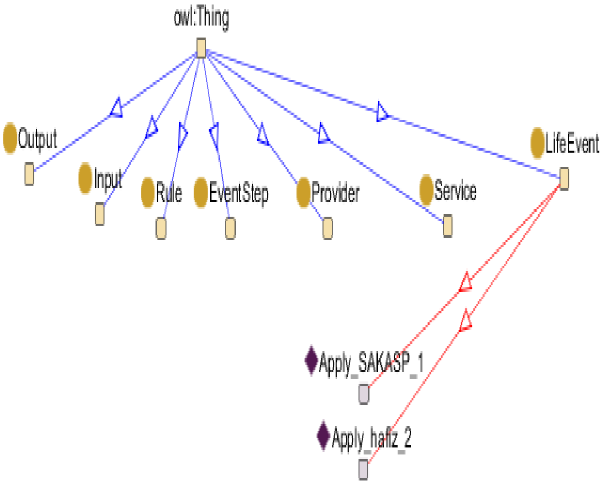
5. RUNNING EXAMPLES FOR TESTING AND EVALUATING THE LIFE EVENT ONTOLOGY BASED E-GOVERNMENT SERVICE

This section presents how we modelled SAKASP and Hafiz as life event examples using our proposed LEO to validate them. Relevant information about the life events, described below, is taken from the SAUDI National e-Government Portal website [11].

As shown in Fig. 9, two life events are defined:

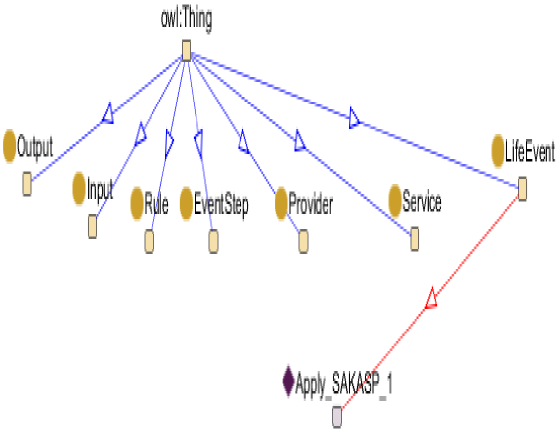
1. Apply SAKASP.
2. Apply Hafiz.

Fig. 9: LIFE EVENT CLASS AND ITS INSTANCES



For the purposes of this study, we will examine 'Apply SAKASP', as shown in Fig. 10.

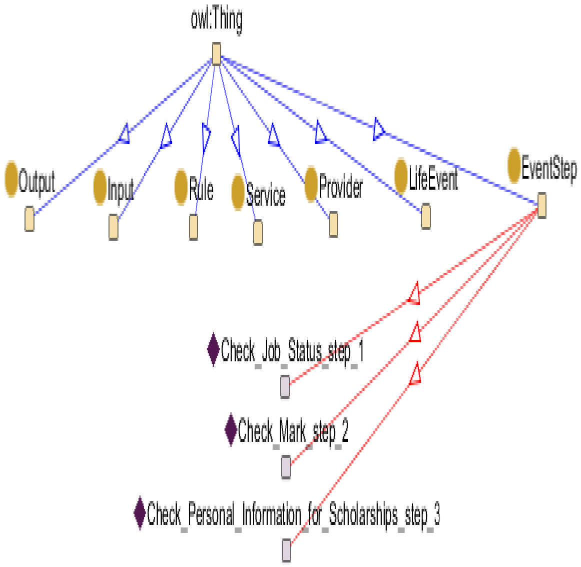
Fig. 10: 'APPLY SAKASP' AS AN INSTANCE OF THE LIFE EVENT CLASS



As depicted in Fig. 11, 'Apply SAKASP' leads the integration of three steps. These steps are relevant to the following:

1. Check Job Status Step 1.
2. Check Mark Step 2.
3. Check Personal Information for Scholarship Step 3.

Fig. 11: INSTANCES OF EVENT STEP CLASS FOR APPLY SAKASP

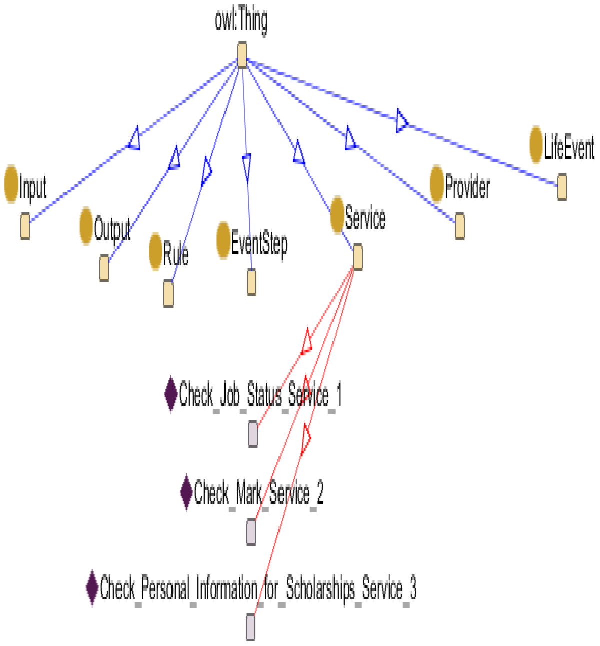


According to our LEO, each event step consumes one service. Therefore, SAKASP's event steps consume:

1. Check Job Status Service. It is consumed by event Step1.
2. Check Mark Service. It is consumed by event Step2.
3. Check Personal Information for Scholarship Service. It is consumed by event Step3.

Those event steps are depicted in Fig. 12.

Fig. 12: INSTANCES OF SERVICE CLASS FOR 'APPLY SAKASP'

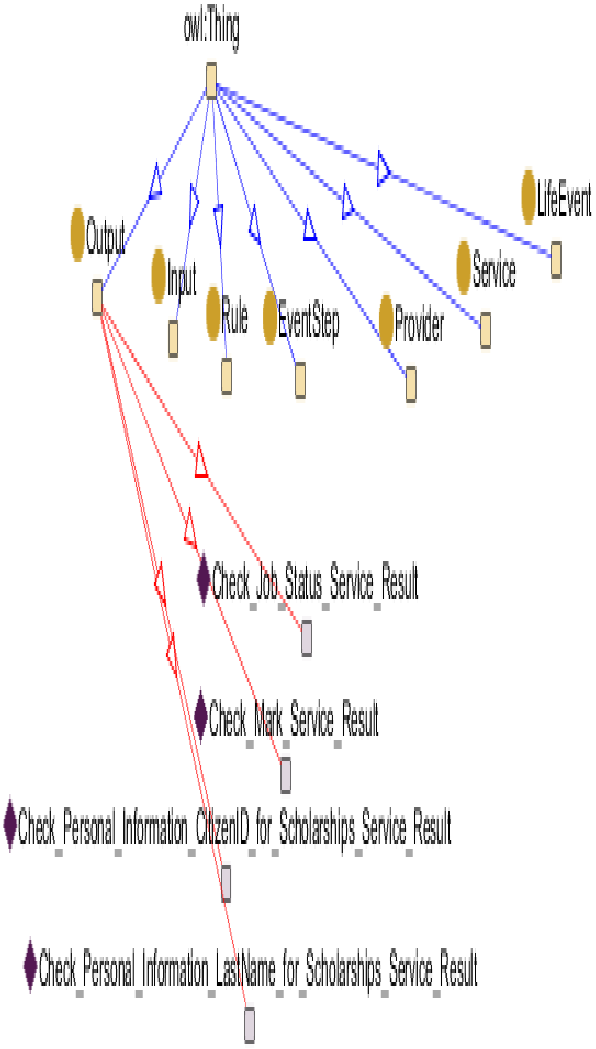

owl:Thing
Input
Output
Rule
EventStep
Service
Provider
LifeEvent
Check_Job_Status_Service_1
Check_Mark_Service_2
Check_Personal_Information_for_Scholarships_Service_3

The services shown above have been designed to produce specific outputs for a particular service. The outputs include the following:

1. The output of the first service is the citizen's job status, which is 'has job' or 'hasn't job'.
2. The output of the second service is the citizen's mark, which is between 0 and 100.
3. The output of the third service is the citizen's personal information (Lastname – CitizenID).

Those outputs are depicted in Fig.13.

Fig. 13: INSTANCES OF OUTPUT CLASS FOR APPLY SAKASP

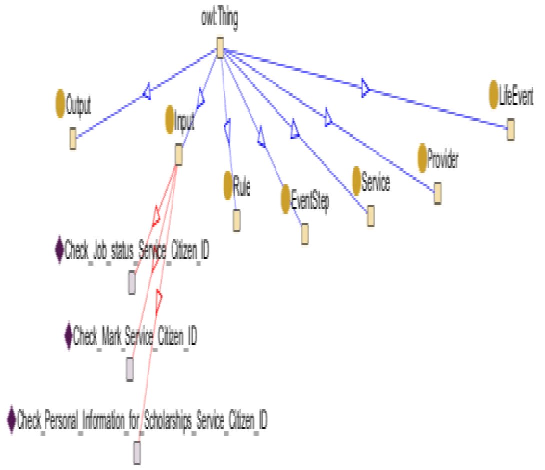

owl:Thing
Output
Input
Rule
EventStep
Provider
Service
LifeEvent
Check_Job_Status_Service_Result
Check_Mark_Service_Result
Check_Personal_Information_CitizenID_for_Scholarships_Service_Result
Check_Personal_Information_LastName_for_Scholarships_Service_Result

Those outputs are based on specific inputs. The inputs are:

1. To achieve the first service, citizen should enter his/her citizen ID.
2. To achieve the second service, citizen should enter his/her citizen ID.
3. To achieve the third service, citizen should enter his/her citizen ID.

Those inputs are depicted in Fig. 14.

Fig. 14: INSTANCES OF INPUT CLASS FOR ‘APPLY SAKASP’

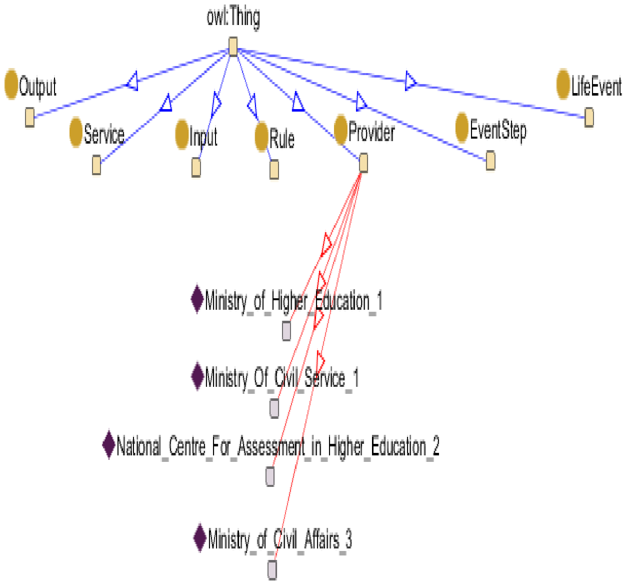

owl:Thing
Output
Input
LifeEvent
Rule
EventStep
Service
Provider
Check_Job_status_Service_Citizen_ID
Check_Mark_Service_Citizen_ID
Check_Personal_Information_for_Scholarships_Service_Citizen_ID

According to the services above and ‘Apply SAKASP’, the providers who are responsible for offering services and life events are:

1. Ministry of Higher Education offers Apply SAKASP.
2. Ministry Of Civil Service offers Check Job Status Service.
3. National Centre for Assessment in Higher Education offers Check Mark Service.
4. Ministry of Civil Affairs offers Check Personal Information for Scholarship Service.

Those providers are depicted in Fig. 15.

Fig. 15: INSTANCES OF PROVIDER CLASS FOR ‘APPLY SAKASP’

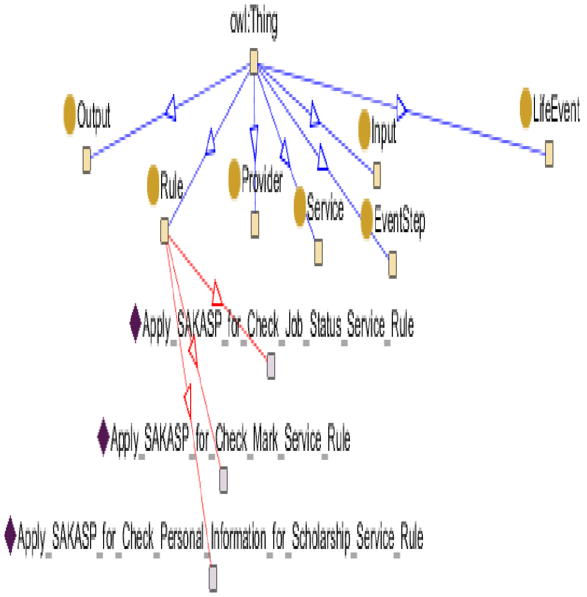

owl:Thing
Output
LifeEvent
Service
Input
Rule
Provider
EventStep
Ministry_of_Higher_Education_1
Ministry_Of_Civil_Service_1
National_Centre_For_Assessment_in_Higher_Education_2
Ministry_of_Civil_Affairs_3

For ‘Apply SAKASP’, there are rules that are implemented on the output. The rules are:

1. Apply SAKASP for Check Job Status Service Rule.
2. Apply SAKASP for Check Mark Service Rule.
3. Apply SAKASP for Check Personal Information for Scholarship Service Rule.

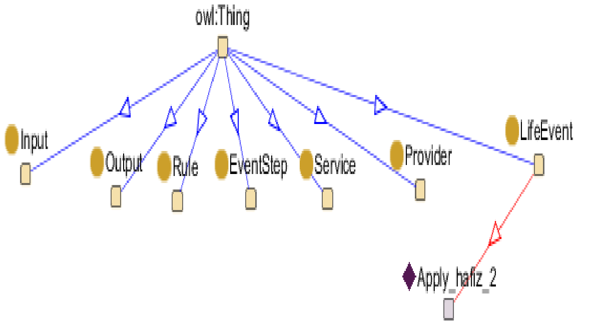
Those rules are depicted in Fig. 16.

Fig. 16: INSTANCES OF RULE CLASS FOR ‘APPLY SAKASP’


owl:Thing
Output
LifeEvent
Input
Rule
Provider
Service
EventStep
Apply_SAKASP_for_Check_Job_Status_Service_Rule
Apply_SAKASP_for_Check_Mark_Service_Rule
Apply_SAKASP_for_Check_Personal_Information_for_Scholarship_Service_Rule

We presented another life event and used our LEO to validate it. This life event is ‘Apply Hafiz’. For the purposes of this study, we will examine ‘Apply Hafiz’ as shown in Fig. 17.

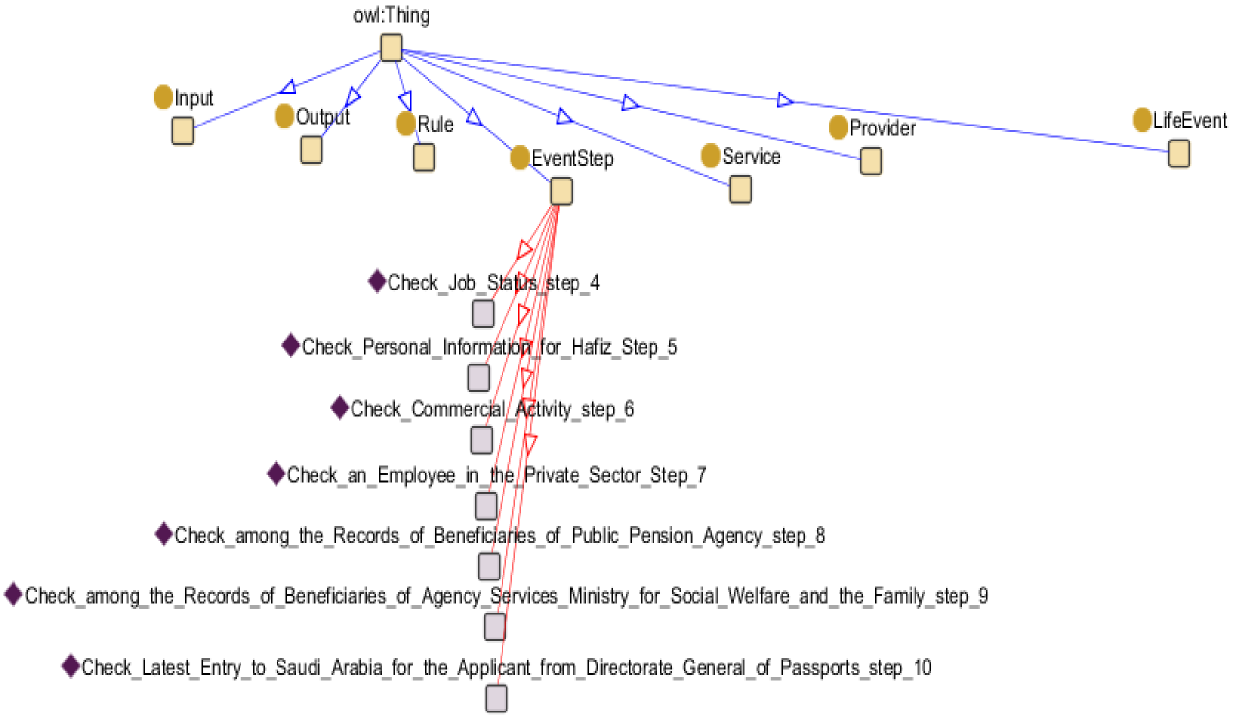
Fig. 17: ‘APPLY HAFIZ’ AS INSTANCE OF LIFE EVENT CLASS



As depicted in Fig. 18, ‘Apply Hafiz’ leads to the integration of three steps. These steps are relevant to the following:

1. Check Job Status Step 4.
2. Check Personal Information for Hafiz Service Step 5.
3. Check Commercial Activity Step 6.
4. Check an Employee in the Private Sector Service Step 7.
5. Check among the Records of Beneficiaries of Public Pension Agency Service Step 8.
6. Check among the Records of Beneficiaries of Agency Services Ministry for Social Welfare and the Family Service Step 9.
7. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service Step 10.

Fig. 18: INSTANCES OF EVENT STEP CLASS FOR ‘APPLY HAFIZ’

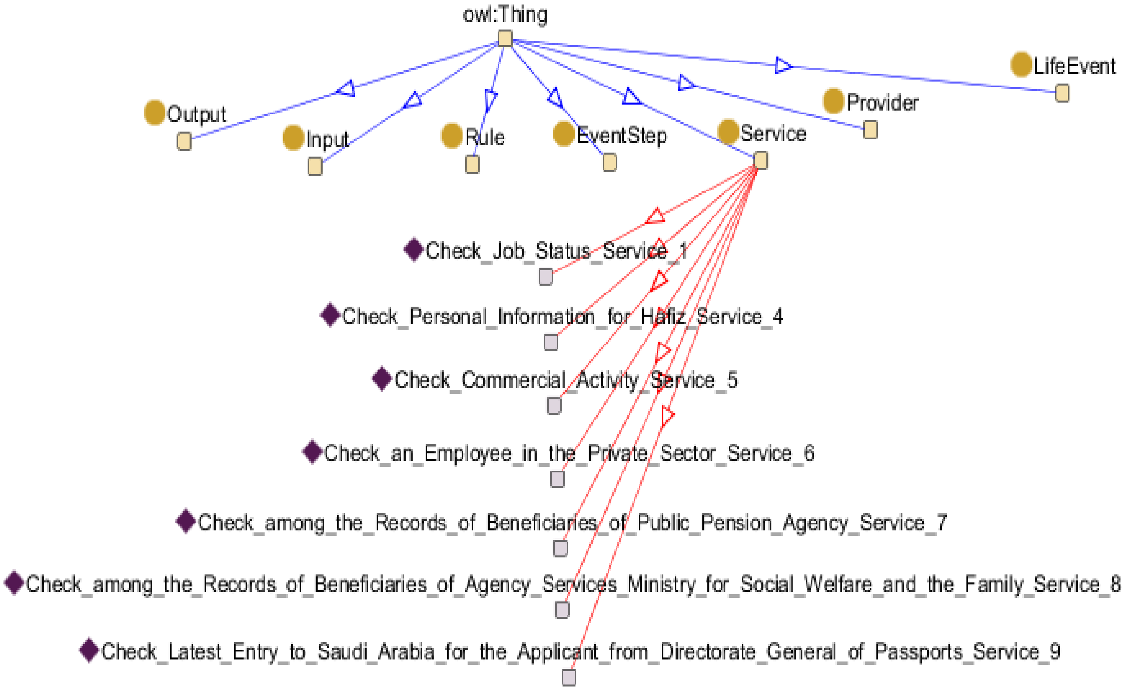

owl:Thing
Input
Output
Rule
EventStep
Service
Provider
LifeEvent
Check_Job_Status_step_4
Check_Personal_Information_for_Hafiz_Step_5
Check_Commercial_Activity_step_6
Check_an_Employee_in_the_Private_Sector_Step_7
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_step_8
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_step_9
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_step_10

According to our LEO, each event step consumes one service. Therefore, Hafiz’s event steps consume:

1. Check Job Status Service.
2. Check Personal Information for Hafiz Service.
3. Check Commercial Activity Service.
4. Check an Employee in the Private Sector Service.
5. Check among the Records of Beneficiaries of Public Pension Agency Service.
6. Check among the Records of Beneficiaries of Agency Services Ministry for Social Welfare and the Family Service.
7. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service.

These outputs are depicted in Fig. 19.

Fig. 19: INSTANCES OF SERVICE CLASS FOR ‘APPLY HAFIZ’

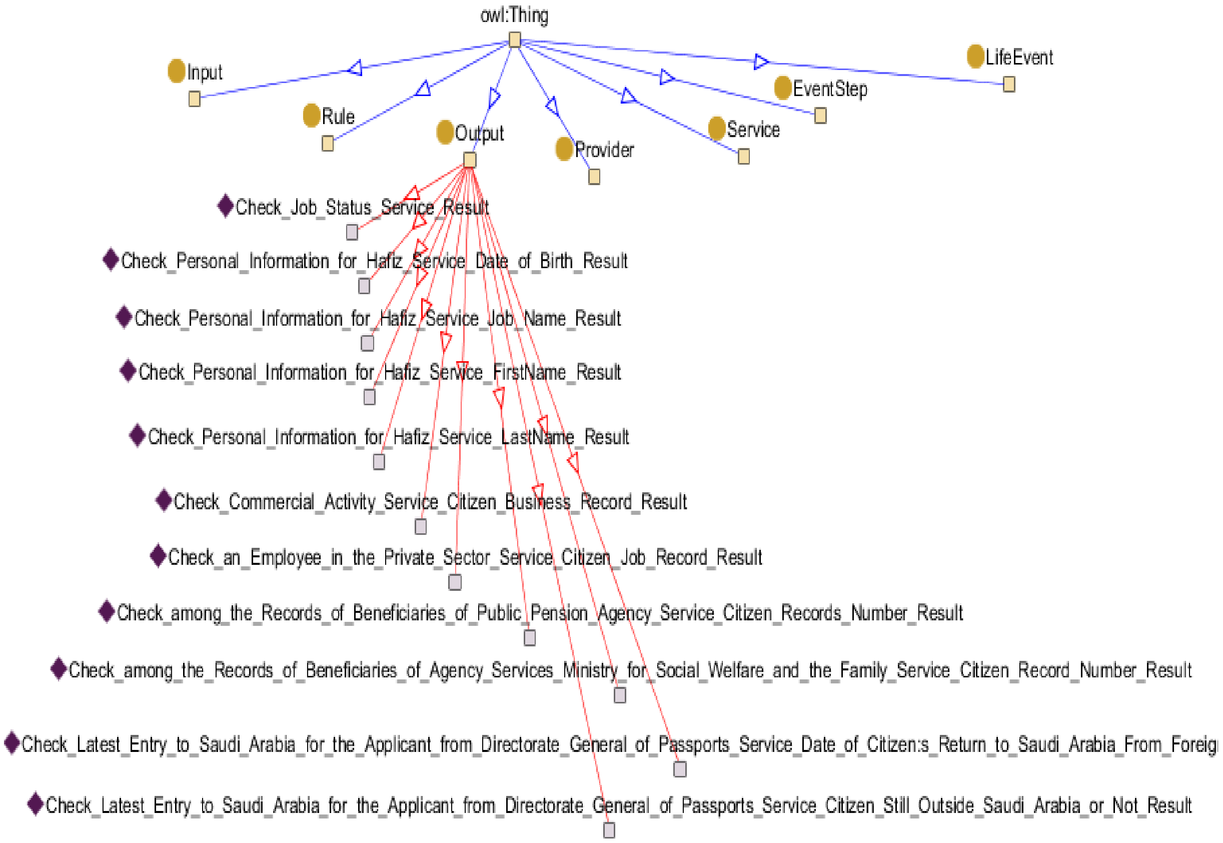

owl:Thing
Output
Input
Rule
EventStep
Service
Provider
LifeEvent
Check_Job_Status_Service_1
Check_Personal_Information_for_Hafiz_Service_4
Check_Commercial_Activity_Service_5
Check_an_Employee_in_the_Private_Sector_Service_6
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_7
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_Service_8
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_9

The services shown above have been designed to produce specific outputs for a particular service. The outputs are:

1. Check Commercial Activity Service Citizen Business Record Result.
2. Check Job Status Service Result.
3. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service Citizen Still Outside Saudi Arabia or Not Result.
4. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service Date of Citizen's Return to Saudi Arabia From Foreign Countries Result.
5. Check Personal Information for Hafiz Service Date of Birth Result.
6. Check Personal Information for Hafiz Service First Name Result.
7. Check Personal Information for Hafiz Service Job Name Result.
8. Check Personal Information for Hafiz Service Last Name Result.
9. Check among the Records of Beneficiaries of Agency Services Ministry for Social Welfare and the Family Service Citizen Record Number Result.
10. Check among the Records of Beneficiaries of Public Pension Agency Service Citizen Records Number Result.
11. Check an Employee in the Private Sector Service Citizen Job Record Result.

These outputs are depicted in Fig. 20.

Fig. 20: INSTANCES OF OUTPUT CLASS FOR APPLY HAFIZ

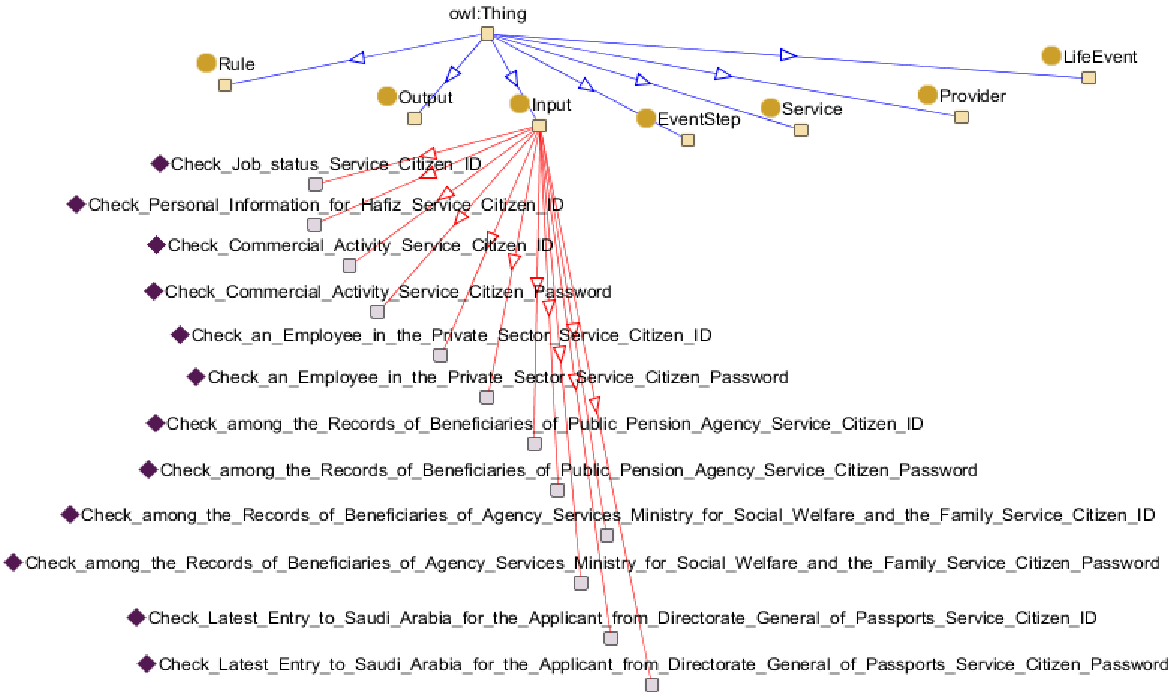

owl:Thing
Input
Rule
Output
Provider
Service
EventStep
LifeEvent
Check_Job_Status_Service_Result
Check_Personal_Information_for_Hafiz_Service_Date_of_Birth_Result
Check_Personal_Information_for_Hafiz_Service_Job_Name_Result
Check_Personal_Information_for_Hafiz_Service_FirstName_Result
Check_Personal_Information_for_Hafiz_Service_LastName_Result
Check_Commercial_Activity_Service_Citizen_Business_Record_Result
Check_an_Employee_in_the_Private_Sector_Service_Citizen_Job_Record_Result
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_Citizen_Records_Number_Result
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_Service_Citizen_Record_Number_Result
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_Date_of_Citizen:s_Return_to_Saudi_Arabia_From_Foreig
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_Citizen_Still_Outside_Saudi_Arabia_or_Not_Result

These outputs are based on specific inputs. The inputs are:

1. Check Commercial Activity Service Citizen ID.
2. Check Commercial Activity Service Citizen Password.
3. Check Job Status Service Citizen ID.
4. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service Citizen ID.
5. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service Citizen Password.
6. Check Personal Information for Hafiz Service Citizen ID.
7. Check among the Records of Beneficiaries of Agency Services Ministry for Social Welfare and the Family Service Citizen ID.
8. Check among the Records of Beneficiaries of Agency Services Ministry for Social Welfare and the Family Service Citizen Password.
9. Check among the Records of Beneficiaries of Public Pension Agency Service Citizen ID.
10. Check among the Records of Beneficiaries of Public Pension Agency Service Citizen Password.
11. Check an Employee in the Private Sector Service Citizen ID.
12. Check an Employee in the Private Sector Service Citizen Password.

These inputs are depicted in Fig. 21.

Fig. 21: INSTANCES OF INPUT CLASS FOR APPLY HAFIZ

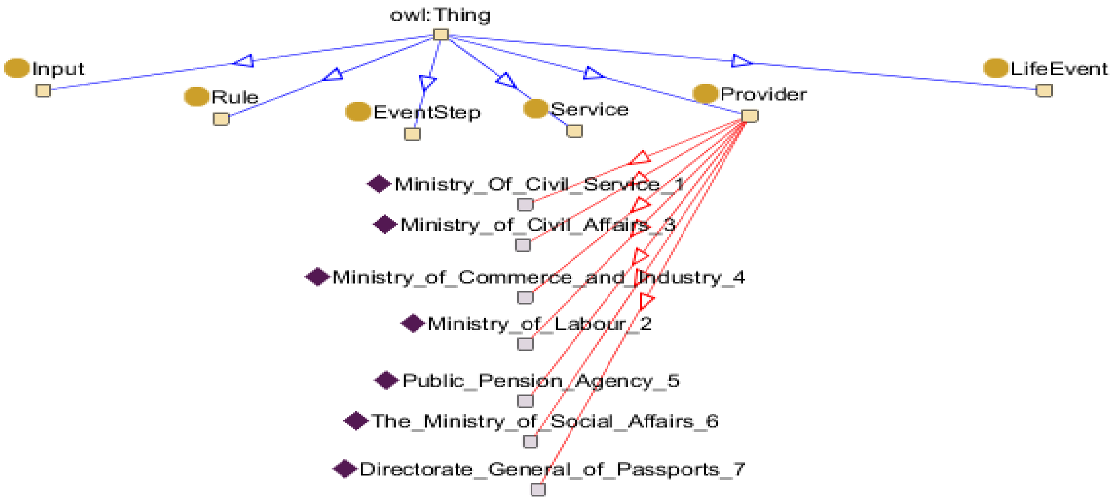

owl:Thing
Rule
Output
Input
EventStep
Service
Provider
LifeEvent
Check_Job_status_Service_Citizen_ID
Check_Personal_Information_for_Hafiz_Service_Citizen_ID
Check_Commercial_Activity_Service_Citizen_ID
Check_Commercial_Activity_Service_Citizen_Password
Check_an_Employee_in_the_Private_Sector_Service_Citizen_ID
Check_an_Employee_in_the_Private_Sector_Service_Citizen_Password
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_Citizen_ID
Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_Citizen_Password
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_Service_Citizen_ID
Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_Service_Citizen_Password
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_Citizen_ID
Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_Citizen_Password

According to the services above and ‘Apply Hafiz’, the providers who are responsible for offering services and life events are:

1. Ministry Of Civil Service.
2. Ministry of Civil Affairs.
3. Ministry of Commerce and Industry.
4. Ministry of Labour.
5. Public Pension Agency.
6. Ministry of Social Affairs.
7. Directorate General of Passports.

These providers are depicted in Fig. 22.

Fig. 22: INSTANCES OF PROVIDER CLASS FOR APPLY HAFIZ

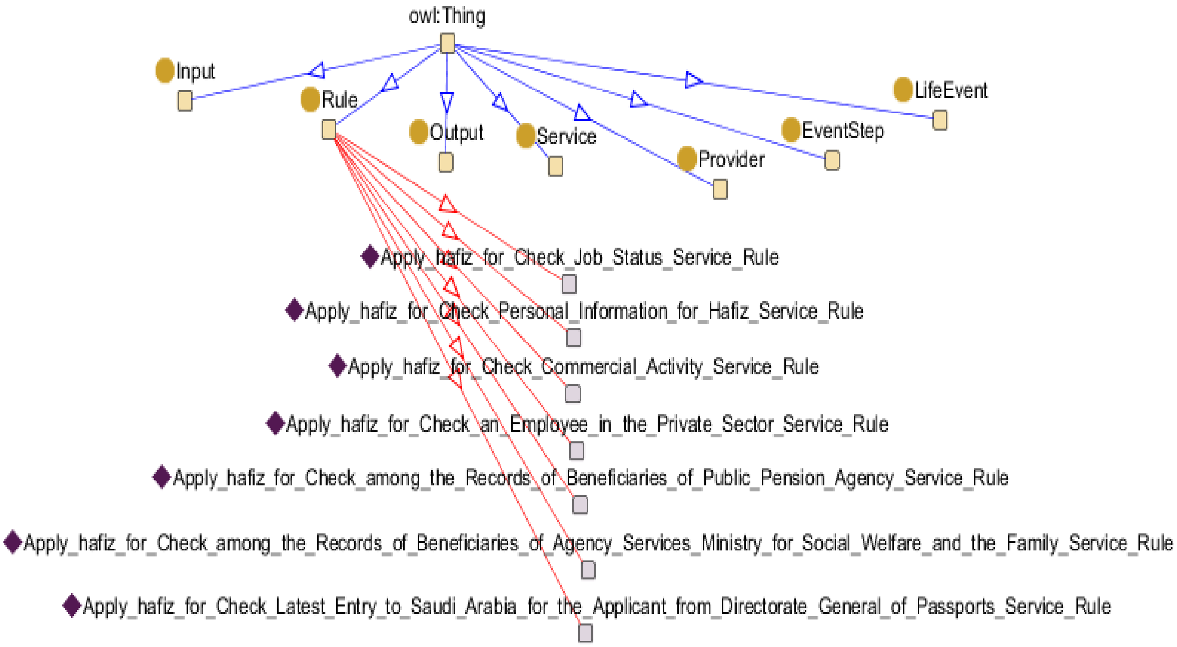

owl:Thing
Input
Rule
EventStep
Service
Provider
LifeEvent
Ministry_Of_Civil_Service_1
Ministry_of_Civil_Affairs_3
Ministry_of_Commerce_and_Industry_4
Ministry_of_Labour_2
Public_Pension_Agency_5
The_Ministry_of_Social_Affairs_6
Directorate_General_of_Passports_7

For ‘Apply Hafiz’, there are rules which implement the input and output. The rules are:

1. Apply Hafiz Rule.
2. Check Job Status Service Rule.
3. Check Personal Information for Hafiz Service Rule.
4. Check Commercial Activity Service Rule.
5. Check an Employee in the Private Sector Service Rule.
6. Check among the Records of Beneficiaries of Public Pension Agency Service Rule.
7. Check among the Records of Beneficiaries of Agency Services Ministry for Social Welfare and the Family Service Rule.
8. Check Latest Entry to Saudi Arabia for the Applicant from Directorate General of Passports Service Rule.

These rules are depicted in Fig. 23.

Fig. 23: INSTANCES OF RULE CLASS FOR 'APPLY HAFIZ'


owl:Thing
Input
Rule
Output
Service
Provider
EventStep
LifeEvent
Apply_hafiz_for_Check_Job_Status_Service_Rule
Apply_hafiz_for_Check_Personal_Information_for_Hafiz_Service_Rule
Apply_hafiz_for_Check_Commercial_Activity_Service_Rule
Apply_hafiz_for_Check_an_Employee_in_the_Private_Sector_Service_Rule
Apply_hafiz_for_Check_among_the_Records_of_Beneficiaries_of_Public_Pension_Agency_Service_Rule
Apply_hafiz_for_Check_among_the_Records_of_Beneficiaries_of_Agency_Services_Ministry_for_Social_Welfare_and_the_Family_Service_Rule
Apply_hafiz_for_Check_Latest_Entry_to_Saudi_Arabia_for_the_Applicant_from_Directorate_General_of_Passports_Service_Rule

6. Discussion and Conclusions

This paper presented an LEO as a tool for supporting e-government service integration. The findings of this study are in line with existing studies, showing that using a semantic web-based ontology has potential benefits in multi-department e-government systems where integration is vital in meeting the needs and expectations of service seekers. This study is different, to some degree, from other studies. In other studies, most of those studies focuses on defining life events' approaches, defining models of life events' levels and features and a conceptual model of LEO-based e-government service integration rather than provides detailed guidelines for developing LEO-based e-government service integration. More specific, our proposed LEO does not include all concepts presented in service ontologies and LEO. Instead, more influential and relevant concepts were selected. In other studies, separately established life event concepts and service concepts make life-event-based e-government service integration systems quite complicated and abstruse. Implementation of an e-government system based on the life cycle ontology therefore appears to have significant benefits with respect to ensuring that client level requirements are reached and allowing the seamless integration of various information sources and services. Therefore, this paper proposed a semantic web-based ontology model for life events for e-government service integration using a methodology. In addition, this paper implemented the model using the ontology modelling tool Protégé and evaluated the model using Pellet Reasoner and the SPARQL query language. The ontology has been developed with a focus on life events in Saudi Arabia: the SAKASP and Hafiz programs. The challenges highlighted in the case study are similar to those faced by e-government systems. This implies that the use of the life cycle ontology and ontology-building guidelines can help improve different e-government system integrations with the aim of improving the availability of information resources and services. This is in line with studies asserting that semantic web-based ontology can be used to improve service integration.

References

[1] A. Alqahtani, H. Lu and J. Lu, Towards semantic-aware and ontology-based e-government service integration: An applicative case study of Saudi Arabia's King Abdullah scholarship program, in: *Advances in Intelligent Decision Technologies*, G. Phillips-Wren, L. C. Jain, K. Nakamatsu and R. J. Howlett, eds., Springer, Heidelberg, 2010, pp. 403–411.

[2] L. G. A. Amoah, Public policy formation in Africa: Toward a grounded ontology, *Administrative Theory & Praxis*, 4(32) (2010), 606-610.

[3] P. Bednar, K. Furdik, M. Kleimann, R. Klischewski, M. Skokan, and S. Ukena, Semantic integration of eGovernment services in Schleswig-Holstein, in: *Electronic Government, LNCS, (vol. 5184)*, M. A. Wimmer, H. J. Scholl, and E. Ferro, eds., SpringerVerlag, Berlin, Heidelberg, 2008, pp. 315-327.

[4] B. S. Berdid and M. Vintar, Simple Life-Events Ontology in SU(M) O-KIF, *Proceedings of the Third International Conference EGOV 2004*, Springer-Verlag, New York Inc, 2004, pp. 112-119.

[5] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American*, May 2001, pp. 28-37.

[6] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima, Building Legal Ontologies with METHONTOLOGY and WebODE. in: *Law and the Semantic Web*, V. R. Benjamins, P. Casanovas, J. Breuker, and A. Gangemi, eds., Springer-Verlag, Berlin, Heidelberg, 2005, pp. 142-157.

[7] G. P. Dias and J. A. Rafael, A simple model and a distributed architecture for realizing one-stop e-

government, *Electronic Commerce Research and Applications*, 6 (1) (2007), 81-90.

[8] D. Fensel, Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin, Heidelberg, 2001.

[9] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, Methontology: from ontological art towards ontological engineering, *Proceedings of the AAAI-97 Spring Symposium Series on Ontological Engineering*, Stanford University, 1997, pp. 33-40.

[10] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web, Springer-Verlag, Berlin, Germany, 2004.

[11] Saudi Government, SAUDI National e-Government Portal, Retrieved on September 2012 from http://www.saudi.gov.sa/wps/portal

[12] M. Gruninger and M. Fox, Methodology for the Design and Evaluation of Ontologies, *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing held in conjunction with IJCAI-95*, Montreal, 1995.

[13] A. Gugliotta, L. Cabral, J. Domingue, V. Roberto, M. Rowlatt, and R. Davies, A Semantic Web Service-based Architecture for the Interoperability of E-government Services, *Proceedings of the Web Information Systems Modeling Workshop (WISM 2005) in conjunction with The 5th International Conference on Web Engineering (ICWE 2005)*, Sydney, Australia, 2005.

[14] G. Kavadias and E. Tambouris, GovML: a markup language for describing public services and life events, in: *Knowledge Management in e-Government*, LNCS (vol. 2645), M. Wimmer, eds., Springer, 2003, pp. 106-115.

[15] I. Kollia, B. Glimm, and I. Horrocks, SPARQL query answering over OWL ontologies, *Proceedings of the Extended Semantic Web Conference*, May 29–June 2, Heraklion, Greece, 2011.

[16] D. Lavbič, I. Lajovic, and M. Krisper, Facilitating information system development with Panoramic view on data, In: *Computer Science and Information Systems*, 7(4) (2010), pp. 737-767.

[17] L. Liuming, Z. Guojin, and C. Jiaxun, An infrastructure for e-government based on semantic Web services, *Proceedings of the IEEE International Conference on Services Computing*, 2004, pp. 483-486.

[18] B. Medjahed, A. Bouguettaya, and M. Ouzzani, Semantic web enabled e-government services, *Proceedings of the 2003 annual national conference on Digital government research*, Boston, 2003.

[19] M. Momotko, W. Izdebski, E. Tambouris, K. Tarabanis, and M. Vintar, An architecture of active life event portals: generic workflow approach, in: *Electronic Government*, LNCS (vol. 4656), Wimmer, M., Scholl, J., Grφnlund, E. eds., Springer, Heidelberg, 2007, pp. 104-115.

[20] E. Mugellini, M. C. Pettenati, O. A. Khaled, and F. Pirri, eGovernment Service Marketplace: Architecture and Implementation, in: *E-Government: Towards Electronic Democracy*, LNAI (vol. 3416), M. Böhlen, J. Gamper, W. Polasek, and M. A. Wimmer, Eds., Springer, Berlin, Heidelberg, 2005, pp. 193-204.

[21] S. Overbeek, M. Janssen, and P. van Bommel, An ontological framework for integrated public service delivery, *Proceedings of the Third International Workshop on Ontology, Conceptualization and Epistemology for Information Systems*, Amsterdam, Netherlands, 2009, pp. 129-135.

[22] V. Peristeras and K. Tarabanis, Reengineering Public Administration through Semantic Technologies and the GEA Domain Ontology, *Proceedings of the AAAI Spring Symposium*, Stanford University, California, USA., 2006.

[23] E. Prud'hommeaux and A. Seaborne, SPARQL Query Language for RDF, Retrieved on Jun 26, 2012 from http://www.w3.org/TR/rdf-sparql-query/

[24] C. Rodriguez Dominguez, K. Benghazi, M. Noguera, M. Bermudez Edo, and J. L. Garrido, Dynamic Ontology-Based Redefinition of Events Intended to Support the Communication of Complex Information in Ubiquitous Computing, *Journal of Network Protocols and Algorithms,* 2(3) (2010), 85-99.

[25] F. Sanati and J. Lu, Life-event modelling framework for e-government integration, *Electronic Government, an International Journal*, 7(2) (2010), 183 – 202.

[26] [26] H. G. Siebert and E. Siebert, SOA and ITIL Integration using Ontologies, EuroCMG, 2007.

[27] R. Studer, S. Grimm, and A. Abecker, Semantic Web Services: Concepts, Technologies, and Applications, Springer-Verlag, New York, Inc. Secaucus, NJ, USA, 2007.

[28] H. N. Talantikite, D. Aissani, and N. Boudjlida, Semantic annotations for web services discovery and composition, *Computer Standards & Interfaces*, 31(9) (2009), 1108-1117.

[29] L. Todorovski, A. Leben, M. Kunstelj, D. Cukjati, and M. Vintar, Methodology for building models of life events for active portals, *Proceedings of the Fifth International EGOV Conference*, Trauner, 2006, pp. 61-68.

[30] I. Trochidis, E. Tambouris, and K. Tarabanis, Identifying Common Workflow Patterns in Life-Events and Business Episodes, *Proceedings of The second International Conference on e-Government*, Pittsburgh, USA, 2006, pp. 234-243.

[31] I. Trochidis, E. Tambouris, and K. Tarabanis, An ontology for modeling life-events, *Proceedings of the IEEE 2007 International Conference on Services Computing*, Salt Lake City, UT, 2007 2007, pp. 719–720.

[32] M. Uschold and M. Gruninger, Ontologies: Principles, methods and applications, *Knowledge engineering review*, 11(2) (1996), 93-136.

[33] M. Vintar, M. Kunstelj, and A. Leben, Delivering better quality public services through life-event portals, *Proceedings of the 10th NISPAcce Annual Conference*, Cracow, Polan, 2002.

[34] D. Vrandecic, Ontology Evaluation, in: *Handbook on Ontologies*, S. Staab and R. Studer, eds., Springer Berlin Heidelberg, 2009, pp. 293-313.

[35] M. A. Wimmer, A European perspective towards online one-stop government: the eGOV project, In: *Electronic Commerce research and Applications*, 1(1) (2002), 92-103.