YAŞAR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

DEVELOPING A METHOD

AND TOOL

FOR HYBRID DATA BASED ACCESS FRAMEWORK

NURİYE YASEMİN ALPARSLAN

THESIS ADVISOR: ASSOC.PROF. DR. MURAT KOMESLİ

COMPUTER ENGINEERING

PRESENTATION DATE: 10.11.2017

BORNOVA / İZMİR
November, 2017

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Jury Members:	**Signature:**
Assoc.Prof.Dr. Murat KOMESLİ Yaşar University	.....
Prof. Dr. Mehmet Cudi OKUR Yaşar University	.....
Assoc. Prof. Dr. Murat Osman ÜNALIR Ege University	.....

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

DEVELOPING A METHOD AND TOOL FOR HYBRID DATA BASED ACCESS FRAMEWORK

Alparslan, Nuriye Yasemin
MSc, Computer Engineering
Advisor: Assoc.Prof.Dr. Murat Komesli
November 2017

Semantic web aims that web contents are understandable from other software, are to be interpreted, usable and sharable. Security of information in semantic web is normally provided by access control mechanism. OBAC (Ontology Based Access Control) is a model for accessing ontologies. RBAC (Role Based Access Control) is a model for accessing relational databases. In this study, HAC (Hybrid Access Control) is aimed to be applied for mapping of relational database and ontology by means both of OBAC and RBAC together. With this proposed method and a tool software, access control of roles and profiles is aimed to be managed in HR (Human Resources) domain. HAC is improved as an access control by using both RBAC and OBAC. A software architect is presented for the access of framework. As a conclusion, roles are managed in HR domain by using hybrid data and got solution to database management.

Key Words: RBAC, OBAC, HAC, Access Framework, Ontology, Human Resources

ÖZ

MELEZ VERİ TABANLI ERİŞİM ÇERÇEVESİ İÇİN BİR YÖNTEM VE ARACIN GELİŞTİRİLMESİ

Alparslan, Nuriye Yasemin
Yüksek Lisans, Bilgisayar Mühendisliği
Danışman: Doç.Dr. Murat Komesli
Kasım 2017

Anlamsal web, web içeriklerinin diğer yazılımlar tarafından anlaşılabilir, yorumlanabilir, kullanılabilir olmasını ve bilginin paylaşılmasını amaçlamaktadır. Anlamsal web'de bilginin güvenliği erişim denetimi ile sağlanmaktadır. OBAC (Ontology Based Access Control – Ontoloji Tabanlı Erişim Denetimi) verinin anlamsal tanımının olduğu ontolojilere erişim denetiminin sağlanmasında kullanılan bir modeldir. RBAC (Role Based Access Control – Rol Tabanlı Erişim Denetimi) ise, rol tabanlı erişim denetimini sağlamaktadır. İlişkisel veritabanlarında RBAC, ontolojilerde ise OBAC kullanılmaktadır. İnsan Kaynakları (İK) uygulamalarında hem RBAC hem de OBAC kullanılmaktadır. Bu çalışmada, her iki yaklaşımın karışımı olan HAC (Hybrid Access Control- Melez Erişim Denetimi) altyapı mimarisi geliştirilmeye çalışılmıştır. İK alanında, OBAC ve RBAC birleşimiyle oluşan melez veri tabanlı bir erişim çerçevesi olarak HAC geliştirilmektedir. Bu çerçeveye erişim için, yöntem ve araç yazılım içeren bir yazılım mimarisi sunulmaktadır. Böylelikle, melez veri ile roller yönetilerek İK alanında veritabanı yönetimine bir çözüm getirilmiş olacaktır.

Anahtar Kelimeler: RBAC, OBAC, HAC, erişim çerçevesi, insan kaynakları, ontoloji

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Assoc. Prof. Dr. Murat Komesli who supports, believes me since undergraduate years until today for his guidance and patience during this study. His advice on both research as well as on my career have been invaluable.

I am also grateful to Assoc. Prof. Dr. Murat Osman Ünalır for his unfailing support and assistance.

I would like to offer my special thanks to Assist. Prof. Dr. Özgü Can for her guidance.

I would like to express my enduring love and thanks to my husband, my best friend Mustafa Alp Alparslan who always support, trust and believe me.

I wish to thank my dear mother Canan Çam and my dear father Remzi Çam who are always supportive, loving and caring to me in every possible way in my life since the day I was born to today.

Words can not express how grateful I am to my mother-in law Asuman Alparslan, father-in-law Haluk Alparslan for all the sacrifices.

I would especially like to thank to LOGO YAZILIM for the tolerance they have shown during my graduate education and my Software Development Manager Nail Diker for his guidance.

Nuriye Yasemin Alparslan

İzmir, 2017

TEXT OF OATH

I declare and honestly confirm that my study, titled "Developing A Method and Tool for Hybrid Data Based Access Framework" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Nuriye Yasemin Alparslan

Signature

November 10, 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

LIST OF ABBREVIATIONS

HR	Human Resources
CV	Curriculum Vitae
IT	Information System
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role Based Access Control
OBAC	Ontology Based Access Control
ABAC	Attribute-Based Access Control
SO-RBAC	Semantic and Ontology-based Role-Based Access Control
SAC	Semantic Access Control
HAC	Hybrid Access Control
SQL	Structured Query Language
PK	Primary Key
FK	Foreign Key
OWL	Web Ontology Language
OWL-DL	OWL Description Logics
RDF	Resource Description Framework
URI	Universal Resource Identifier

SWRL	Semantic Web Rule Language
SQWRL	Semantic Query-enhanced Web Rule Language
SPARQL	SPARQL Protocol and RDF Query Language
URA	User-Role Assignment (RBAC)
PRA	Role-Permission Assignment (RBAC)

CHAPTER 1
INTRODUCTION

This thesis strives to improve a technique and a tool for the access control of hybrid data containing relational database and ontology in human resources domain. Unlike most works that develop access control to known approaches, or techniques, the emphasis here is to thinking RBAC(Role Based Access Control) and OBAC(Ontology Based Access Control) together in a specific work area in real life- Human Resources.

Human resources covers employees and candidates as a role and interested in these roles all business with their company such as educations, exams, recruitments, competency management, organizational assignments, absentee records and managements, their salaries, etc. Work force management and resource management should be considered together for organizational performance improvements. Accessing and sharing the knowledge about recruitment area that are dispersed semantically is provided greatly after realisation of web technologies by HR specialists. HR ontology is generated as a common library in this area. The requirement of usage and allocation of employee's and candidate's detailed information between companies bring standardization of data. HR ontology provides standardization of HR knowledge. In accordance to this standardization access control and access framework of knowledge become an important thing for security of data.

This study, accordingly, required a appreciably self learning in access control, relational owl, modeling access control with hybrid data, mapping of relational database of HR and ontology. In this work, SQL database has been developed including recruitment information (candidate's CV details), organizational informations (assignment and discharge) and general information (education, competency, etc.) of employees. After mapping of relational database and ontology, access control of the hybrid data is aimed to be done by using both OBAC and RBAC together with a new method and implementation tool software.

An architecture offer is presented for hybrid access control which is formed by combination of OBAC (which is an ontology databased access control enabling semantic integrity at accessing data) and RBAC (which is role based access control as associative model). It will enable to administer the roles and show solution architect

brought to the database administration. In the first part a large extend information is presented about studies and researches made in this subject. In the second part, mapping between RDB and ontology is told. In the third part a construction within human resources scenario for hybrid access is presented. At the last part results and offers related to the issue take place.

1.1. Semantic Web Applications In Human Resources Area

Semantic web provides a common language by using both data and rules for knowledge representation. Because universality is an important property. Machines can conceive semantic documents and data in each other. For expressing the meaning, triple sets are used and each triple is occurred from subject, verb and object. Using a link on a web page by URI (Universal Resource Identifier), subject and object which are triple parts are identified. Every concept is simplified and unified by a URI and anyone can understand these new concepts with less energy. On the web, new concept, a new verb can be defined by URIs. Semantic web helps the knowledge representation, answering a request, to express new concepts anyone in a simplified way. Ontologies that relates the information, improves the web searches by increasing functionality [1].

Human resources ought to be used efficiently with the labor force and resource planning for the improvement of the companies. After distinguishing web technologies as a communication channel by people who are working in HR area, access and allocation of distributed data is provided largely in recruitment processes. Later, HR ontology is created as a common vocabulary [2].

Usage and allocation of candidate and employee information between inter institutions in recruitment processes causes the requirements of standardization of some definitions in HR area. In accordance to this requirement, XML- Schema definitions is made different from each other for the sharing and common definitions of HR data [3].

HR Open Standards are improved by HR professionals and technology specialists. In this way, data integration is provided by developing new standards in HR area [4].

Security of data access that is shared and understood from machines should be provided in accordance to these standards. Security in information system is ensured

by the Access Control. Access framework restricts that any user can access to any resource with any access right. Reviewing of the user rights, managing of accessing to the source with permission of insert, delete, update provides data and source security. Reusage and allocation of data is aimed by semantic web for machines to communicate with each other [1].

Semantic web technologies bring up the security and privacy topics for reusage, allocation and integrity of data [5]. It can be done to limit the access through the source by different mechanisms in access control. By RBAC, authority and permission to the source is given not to the user itself but roles. So when a user is afiliated with a role he/she will have the permission and authority given to those roles. [6].

By OBAC high level data is used at access to the source. Policies are being made up semantically between the subject who would access to the source and high level data [5].

As being a more powerful form of information demonstration, ontologies explain the relations and limitations between definitions. Ontologies made up for human resource sphere include definitions like competency, job perception and their relations. Employment processes and candidates offers became being realized by semantic web technologies. HRXML sign language developed by HR-XML Consortium include more than 75 XML schema independent from each other for human resources ontology [7]. Some semantic web context studies in human resources is explained at further parts.

1.1.1. ECCO System Application

Ecco Project was developed by Italian industry partners AICA and Federcomin and Milan Technical University with support of Italian government. It was aimed to provide guideness and information, transparancy regarding talent and job profiles, comparibility. Ecco Project started with the analysis of national and international ICT (Information and Communication Technologies) talent and job profile approaches. In Ecco Project, there are definitions like knowledge for talent definition, knowledge object, skill that means being able to do something, talent belonging to a specific brunch, performance and job profile[8]. It created a data model by standards it defined

to explain talent definitions, give reference and enable it to be used again[9].

1.1.2. Kowien

Kowien (Kooperatives Wissensmanagement in Engineering-Netzwerken/ Cooperative Knowledge Management in Engineering Networks) is a common research Project at data based system area. Kowien makes ontology based data system defined, developed and evaluated to determine the places of employers in organization according to their talent. To achieve this goal all the related data of the organization is used for establishing a data bas efor employers talents [10]. Kowien is a talent ontology in which the talent definitions collected and defined for determining job position requirements and talents coming at job applications [11].

1.2. Access Control Mechanisms

Mechanisms developed before and after semantic web are as follows:

1.2.1. DAC (Discretionary Access Control)

In DAC policy users give authority and permission to other users. So an access control is user centered. In this way, access control over distributed data can be administered. User who have right to access to one or more sources can give authority and permission to one or more subject for accessing this source [12].

DAC Access model's advantages comparing to other Access models are flexibility, easiness in application. Its disadvantages are distribution of communication authorities being uncontrolled and insecure [5].

1.2.2. MAC (Mandatory Access Control)

MAC controls source access by using access control policy determined by a central authority. The authority central in system is responsible with classifying subjects and sources according to specific security levels. Subjects who have same access level or more can reach to the source [12]. MAC access is realised by hyerarchical control construction. It is generally used at defence sector [13]. While users have authority to write for the sources taking place at a less security level than they have, they normally

have reading authority for more ones [14].

1.2.3. ABAC (Attribute Based Access Control)

ABAC is designed for distributed systems for which subject can't be known before presenting with a request. ABAC gives access right to access some sources and denies for some by taking subject properties as base [12].

ABAC model provides both of MAC and DAC necessities by taking subject, object and their own properties as base. Three especialities are important in ABC model:

Subject properties: Each subject is an existing who can move on a source. Each subject is connected with propertities that constitutes its characteristical and identical information.

Source properties: Source is a place on which a subject can act. Web services, date constructions, system tools can be source.

Environment properties: It can be defined as technical and operational constructions at data access [15].

1.2.4. RBAC (Role Based Access Control)

RBAC controls source access according to the roles of users taking place at a specific group and a common responsibility or duty they have. After appointing appropriate roles for the users access to the same source can be by users having different roles [16]. It allows access management and according to which resource permition can be and does by roles configuration mostly than individual subject constructions. Role based acces control presents the ability to determine subject roles compaing to individual subject definitions. Evaluating access can be controlled in a better way[17].

Role : Context on which permission values and holders attribute. Role context draws the lines for source to act in permission. Roles include permission concept but not other elements.

Permission : Concept means the allowance given by a role. Resources and actions are being elements subjects allowed. That also refers to conditions like period.

Role assignment : Role assignment or it is the reference for which subjects will be exposed to which authority. It draws limitation lines for combinations of roles or how many for which subject will be applied. Role Enablement Authority has that policy whereas it is optional [17].

Each role can have right to process one or more right. Three basic rules are in application:

- Role appoinment: Each subject can make process according to the roles appointed to itself.
- Role authorization: Active role subject has should be authorised for subject.
- Process authorization: Related to the process authority affiliated with subject's active role subject can process [18].

For example, regarding human resources system the roles that can access to the human resources data can be thought as experts and authorities working at human resources department, employers working in all the organization and candidates for job applications. If user is in an human resources specialist role then he can access to the education, certificate info of employers. If user is in a employer role he only sees his own information but can't see someone else's.

1.2.5. OBAC (Ontology Based Access Control)

OBAC model provides access control by semantic web based approach. Policies are established IAW the source and object up data based. The parts that are able to access to source and can be counted as subject, object as being the source and attributes as policy subjects. Policy subjects at OBAC are the deontic subjects explained below:
Permission:: What the object can make.
Prohibition: What object can't make.
Obligatory: What object must do.
Special permission: What object can make in a specific time [19].
There is a study which is made for ontology based access control of semantic web services, usage of ontology provided reasoning talent for access control and decision-making mechanism. It enables to search and find access control information automatically [20].

1.2.5. ROWLBAC (Role Based Access Control in OWL)

New policy languages were made up in access lately. XACML is one of those languages [21]. XACML(eXtensible Access Control Markup Language) is an access control policy language. It provides syntax in XML language for users called subjects or sources and defining demands called movements [22].

New policy languages were developed like Ponder in application and more practical languages resulted from academic studies. They studied on semantic web based Rei and Kaos languages theorically [21].

They include security policies like Ponder. It is an object oriented language for security and administration policies. Rei includes policy rules like authority, obligatory, special permission and gives permission for these policies to be differed to movement, control and policy objects [23].

In ROWLBAC (RBAC in OWL) policy, OWL web ontology language and RBAC are used together. Subject, movement and object definitions at RBAC are being modelised by OWL language [22]. OWL (Web Ontology Language is a semantic markup language used for sharing and broadcasting ontologies on www. It is an enlarged version of RDF (Resource Description Framework) dictionary and made up from DAML+OIL web ontology [24].

ROWLBAC rules can be signified at SWRL Semantic Web Rule Language or N3 (Notation 3). ROWLBAC supports authorisation policy [25].

CHAPTER 2
MAPPING HR RELATIONAL DATABASE TO HR ONTOLOGY

2.1 Relational Database Details

In this study, there is a HR Application database and database contains employee's detailed information, employee's organizational information, candidate's detailed CV information, competency requirement of a position and employee's competency level, user information to login application, user right information for modules and module's programs. Database name is THESIS. Database contains 40 tables with 634 columns. Ontology is designed in Protégé.Table names and their conceptual contents explained in Table 1.

Table 1. Table Names and their conceptual contents

Table Name	Contents
KADEME	Grades of positions
POZISYONYETKINLIK	Competency levels of positions
POZISYON	Organizational position names, values, levels
KULLANP	User names, user ids of applications
TAHSIL	Employee's educational information such as high school, university, master etc.
KULLANHAKP	Contains user rights for modules
CVADAY	Contains candidate's CV details.
OZLUK	Employees detailed information
KULLANPROGP	User rights for module's programs
CV	CV information of candidates
AUTHORIZATIONGROUP	Authorization group.
FIRMA	Company information in detail.

AUTHORIZATIONGROUPOBJECT	Authorization information with group and organizational unit information (sub region, section, department, position)
SICILGECMIS	Organizational history in case of transfers between firms on a company.
AUTHORIZATIONGROUPUSER	Contains relation between users and authorization groups.
CVILAN	Advertisements for recruitments
PROFIL	Contains relation between employees and competencies.
ISLETMELER	Company information.
CVEHLIYET	Candidate's driver license information.
EHLIYET	Driver license information of employees
ILETISIM	Communication information of employees
POZISYONPROFILTASARIM	Relation between position and competencies. A position's requirements are taken.
POZISYONPROFILTASARIMSONUCLAR	Employee list who are suitable to a position according to its position's requirements
CVILANGRUP	Recruitment advertisement group definitions.
CVILANYAYIN	Advertisement publish information
NITELIK	Competency definitions
YETKINLIK	Competency level definitions
CVILAN_IL	Advertisement city information
ULKE	Country definitions
IL	City definitions
SICIL	Start and finish dates of employees
ILCE	Hometown definitions
FAKULTE	Faculty definitions
OBJETIPI	Object definitions of applications
ALTBOLGE	Sub region definitions
CVTAHSIL	Candidate's educational information
KULLANICI	User information of candidates
BOLUM	Organizational section information
DEPARTMAN	Organizational department information
ORGGECMIS	Organizational history

Employees' organizational information are retrieved from tables: ISLETMELER, KADEME, ALTBOLGE, BOLUM, DEPARTMAN, POZISYON, ORGGECMIS, SICIL, SICILGECMIS, FIRMA.

General tables that is used by other tables are: ULKE, IL, ILCE, FAKULTE

User information are taken in tables: AUTHORIZATIONGROUP, AUTHORIZATIONGROUPOBJECT, AUTHORIZATIONGROUPUSER, KULLANPROGP, KULLANICI, KULLANP, KULLANHAKP

Recruitment and candidate information are taken from tables: CV, CVADAY, CVILAN, CVILAN_IL, CVILANGRUP, CVILANYAYIN, CVTAHSIL.

Employees' general information are taken from tables: ILETISIM, NITELIK, OZLUK, TAHSIL, YETKINLIK

Position information is taken from tables: POZISYON, POZISYONPROFILTASARIM, POZISYONPROFILTASARIMSONUCLAR.

2.2 Ontology Design

There are many studies about converting relational database to ontology. A tool software is developed for mapping between relational database data to ontology. Generally the tool is developed in Java using Jena as this study do. In this thesis, mapping tool is developed in C# using JenaDotNet [26].

Some mapping rules are needed for modeling metadata and data transformation from relational database to owl. Rule are related with mapping tables, columns, constraints, rows [27]. The representation of relational data and schema with the Web Ontology Language OWL in this study is made by Protégé as in the Figure 1. With this approach, unnecessary data usage would be prevented.

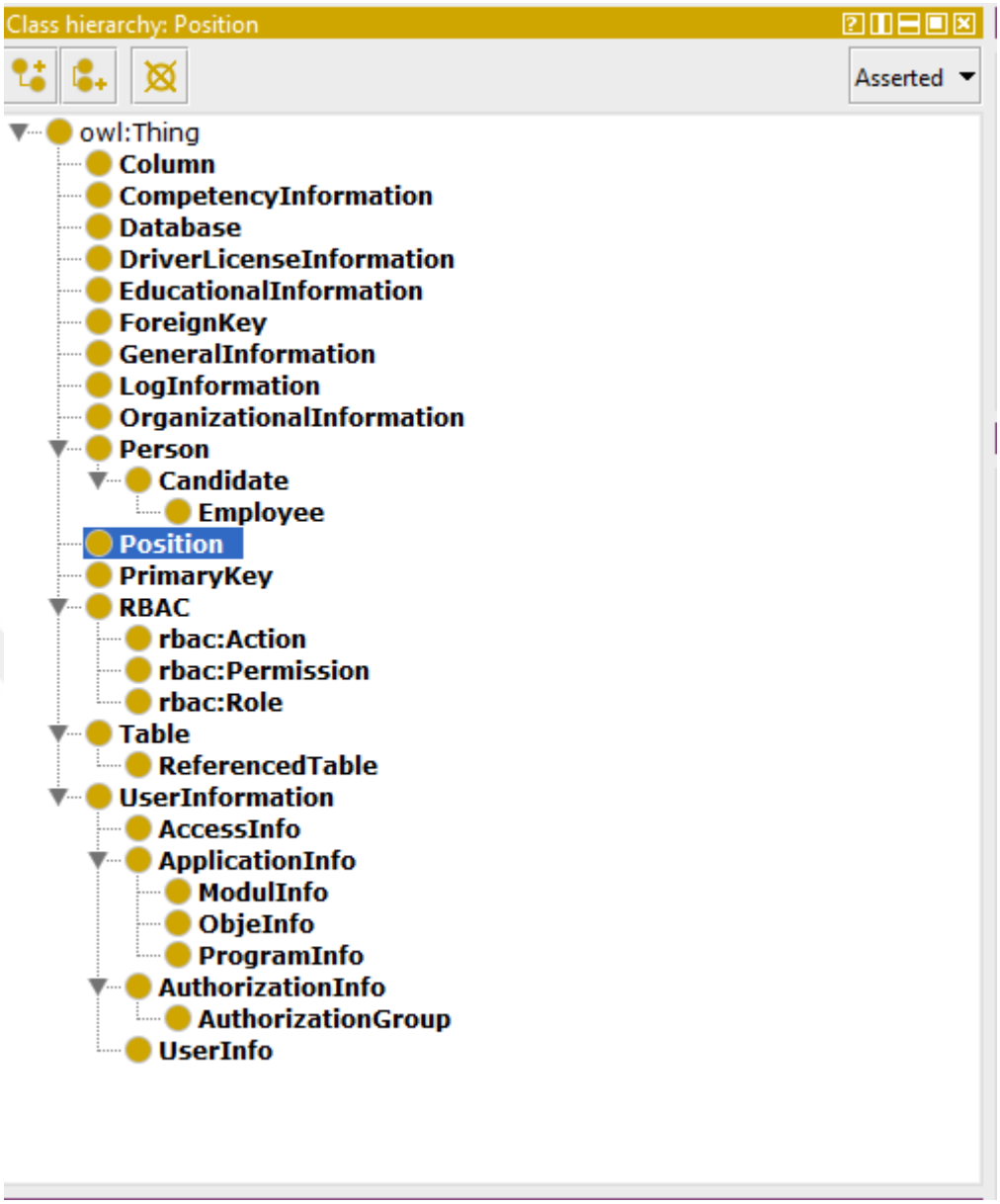


Figure 1. Classes of ontology

The metadata items that I have included in relational owl ontology are database, table, column, primary key, foreign key. As it is shown in Figure 2, relational schema class names and their meanings are explained below.

- Column: Database column names are referred. (*see Appendix 2*)
- Database: Database name is referred.
- ForeignKey: Foreign key names are referred (*see Appendix 3*). The critical point is FK does not correspond to a column. Because FK can consist a column

or a column set.

- PrimaryKey: Primary key names are referred (*see Appendix 3*). The critical point is PK does not correspond to a column. Because PK can consist a column or a column set.

- Table: All table names in THESIS database is defined as individual in Table class.

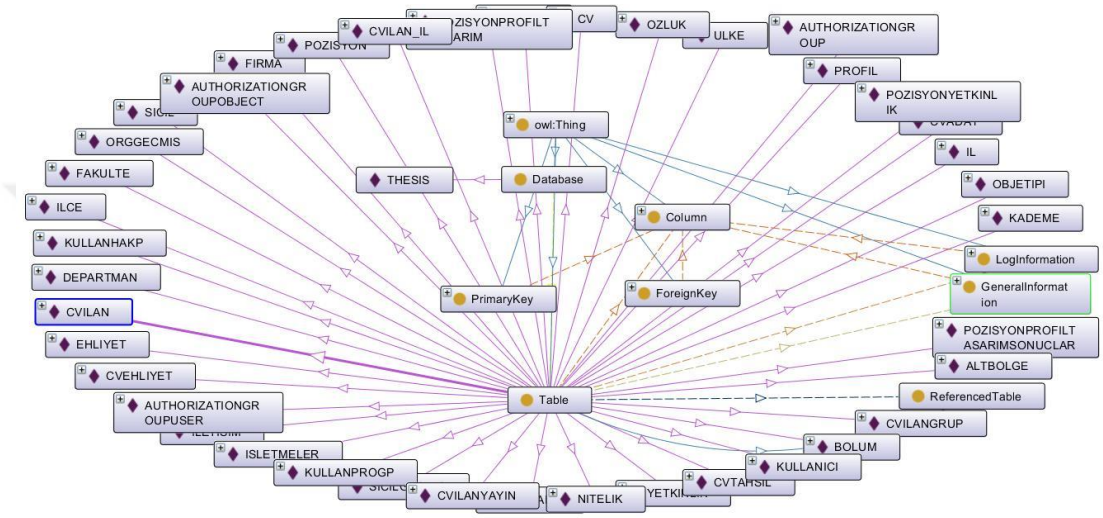


Figure 2. Metadata items in ontology design

It is observed that some column names repeated in different tables with the same meaning *(see Appendix 2)*. Also, some same column set is used in different tables.

While creating the ontology, conceptual grouping of the columns should be seen and performed in to an ontology. Considering the conditions of use and the relational database, some rules can be emphasized specifically.

Knowledge representation of the relational database is provided with the properties in Table 2.

Table 2. Object properties defined in the Relational OWL ontology

Object Property	rdfs:Domain	rdfs:Range	rdfs:Comment
hasTable	Database	Table	A database has a set of tables
hasColumn	Table	Column	A table has a set of columns
	PrimaryKey		A primary key has one or more than one columns
	ForeignKey		A foreign key has one or more than one columns
	DriverLicenseInformation		A driver licence information has one or more than one columns
	GeneralInformation		A general information has one or more than one columns
	CompetencyInformation		A competency information has one or more than one columns
	UserInformation		An user information has one or more than one columns
	OrganizationalInformation		An organizational information has one or more than one columns
	EducationalInformation		An educational information has one or more than one columns
	LogInformation		A log information has one or more than one columns
	Person		A person has one or more than one columns
hasPK	Table	PrimaryKey	A table has one or more than one primary key
hasFK	Table	ForeignKey	A table has one or more than one foreign key
references	Table	ReferencedTable	Table references ReferencedTable
hasAuthorizationInfo	Employee	AuthorizationInfo	An employee can have an authorizationInfo
hasUserInfo	Employee	UserInfo	An employee can have an userinfo
hasPosition	Person	Position	A person can work on a position
hasApplicationAccess	Employee	AccessInfo	An employee has one or more than one accessinfo
hasDriverLicence	Table	DriverLicenceInformation	A table has DriverLicenceInformation
	Employee		An employee can have a driverlicenceinformation
	Candidate		A candidate can have a driverlicenceinformation
hasEducationalInfo	Table	EducationalInformation	A table has EducationalInformation
	Employee		An employee can have a EducationalInformation
	Candidate		A candidate can have a EducationalInformation
hasGeneralInfo	Table	GeneralInformation	A table has GeneralInformation
	Employee		An employee can have a GeneralInformation
	Candidate		A candidate can have a GeneralInformation
hasLog	Table	LogInformation	A table has LogInformation
hasOrganizationalInfo	Table	OrganizationalInformation	A table has OrganizastionalInformation
	Employee		An employee can have a OrganizastionalInformation
	Candidate		A candidate can have a OrganizastionalInformation
hasCompetency	Person	CompetencyInformation	A person has one or more than one competency information
	Position	CompetencyInformation	A position has one or more than one competency level

Since the primary key itself may contain one or more than one column, primary key has “hasColumn” object property.

2.2.1 Conceptual Grouping In Mapping

Ontology is designed by grouping the columns according to their conceptual meanings.

2.2.1.1 Log Information

Every table has “KAYITYAPANKUL”, “KAYITTARIHI”, “DUZELTMEYAPANKUL”, “DUZELTMETARIHI” columns. These columns contains log information. So each table (domain) hasLog (Object Property) LOGINFO(column set of KAYITYAPANKUL, KAYITTARIHI, DUZELTMEYAPANKUL, DUZELTMETARIHI).

Ontology graph of the LogInformation is shown in Figure 3.

- Column name of KAYITYAPANKUL means the username who records the data.
- Column name of DUZELTMEYAPANKUL means the username who edits the data.
- Column name of KAYITTARIHI means the date time of record.
- Column name of DUZELTMETARIHI means the date time of edit.

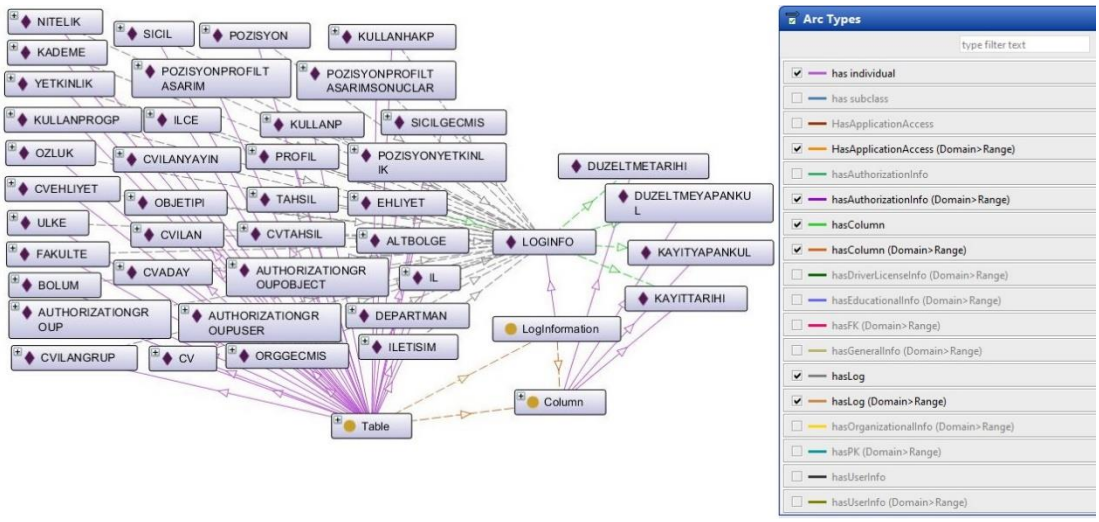
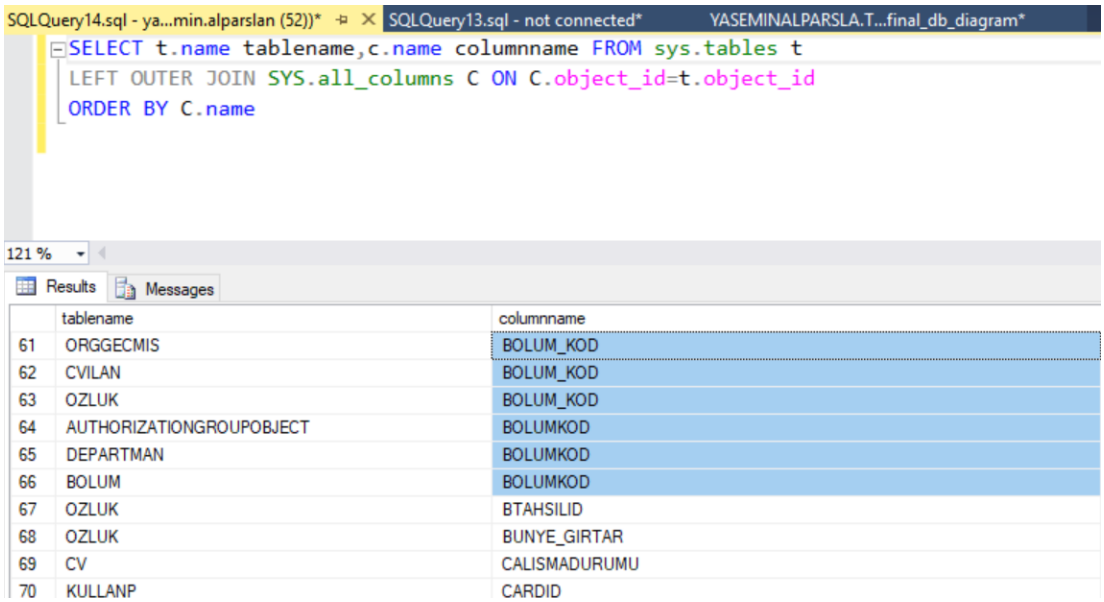


Figure 3. Ontology graph of Log Information

2.2.1.2 Organizational Information

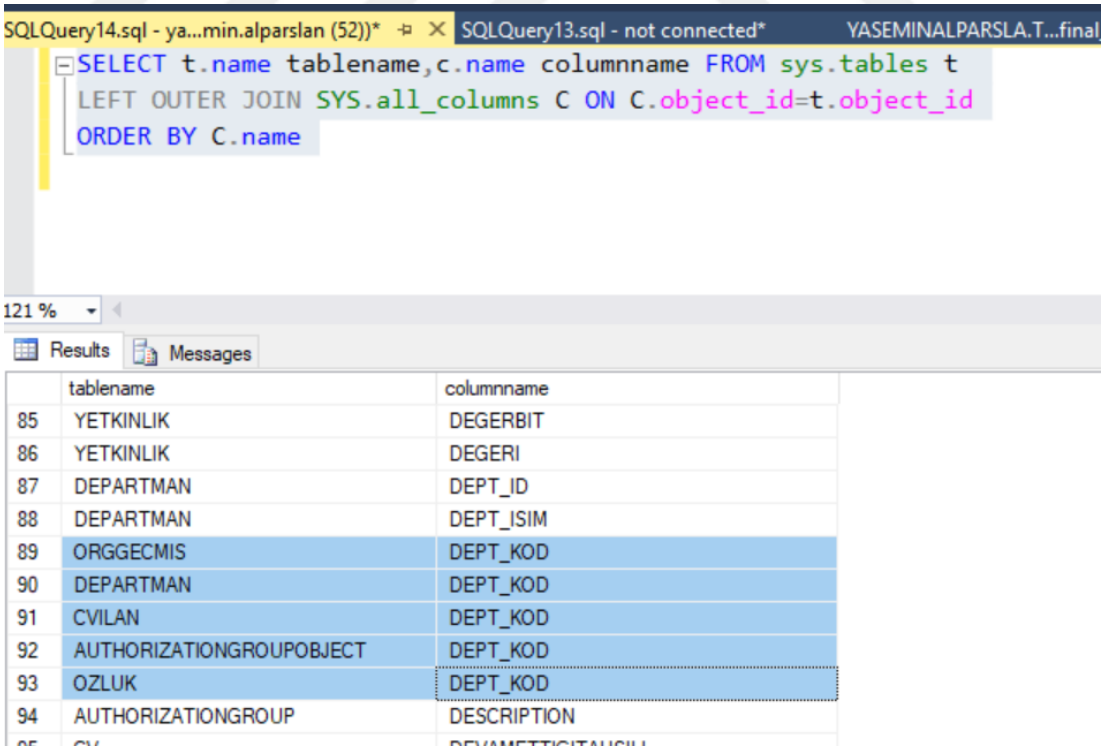
BOLUM_KOD is an organizational information of an employee. As it is shown in Figure 1, BOLUM_KOD column is used in ORGGECMIS, CVILAN, OZLUK, AUTHORIZATIONGROUPOBJECT, DEPARTMAN, BOLUM tables. Tables that have BOLUMKOD and DEPT_KOD are shown in Figure 4 - 5.



	tablename	columnname
61	ORGGECMIS	BOLUM_KOD
62	CVILAN	BOLUM_KOD
63	OZLUK	BOLUM_KOD
64	AUTHORIZATIONGROUPOBJECT	BOLUMKOD
65	DEPARTMAN	BOLUMKOD
66	BOLUM	BOLUMKOD
67	OZLUK	BTAHSILID
68	OZLUK	BUNYE_GIRTAR
69	CV	CALISMADURUMU
70	KULLANP	CARDID

Figure 4. Table list that has BOLUMKOD column

For example ; DEPT_KOD is a organizational information of an employee and is used in ORGGECMIS, DEPARTMAN, AUTHORIZASTIONGROUPOBJECT, OZLUK, CVILAN tables.



	tablename	columnname
85	YETKINLIK	DEGERBIT
86	YETKINLIK	DEGERI
87	DEPARTMAN	DEPT_ID
88	DEPARTMAN	DEPT_ISIM
89	ORGGECMIS	DEPT_KOD
90	DEPARTMAN	DEPT_KOD
91	CVILAN	DEPT_KOD
92	AUTHORIZATIONGROUPOBJECT	DEPT_KOD
93	OZLUK	DEPT_KOD
94	AUTHORIZATIONGROUP	DESCRIPTION

Figure 5. Table list that has DEPT_KOD column

DEPT_KOD and BOLUM_KOD is an organizational information of an employee in this database and these columns are used repeatedly with the same meaning. The meaning is organizational information like DEPT_KOD and BOLUM_KOD, organizational information is mentioned with also POZISYON_KOD and ALTBOLGE_KOD.

In ontology, “OrganizationalInfo” class should be created and organizatioanlinfo class has column
“DEPT_KOD”, ”BOLUM_KOD”, ”ALTBOLGE_KOD”, ”POZISYON_KOD”.

And different table contains these column names together so, some tables can have an organizational information. “hasOrganizationalinfo “object property should be created.

- A table (domain) hasOrganizationalinfo (objectproperty) organizationalinfo (range).

  In Protégé, under the Thing class, OrganizationalInformation class is created as shown in Figure 6. An individual called “ORGANIZATIONALINFORMATION” is created and DEPT_KOD, POZISYON_KOD, ALTBOLGE_KOD, BOLUM_KOD columns joined with hasColumn object property.

  Ontology class of the OrganizationalInformation is shown in Figure 6 and ontology graph of OrganizationalInformation is shown in Figure 7.

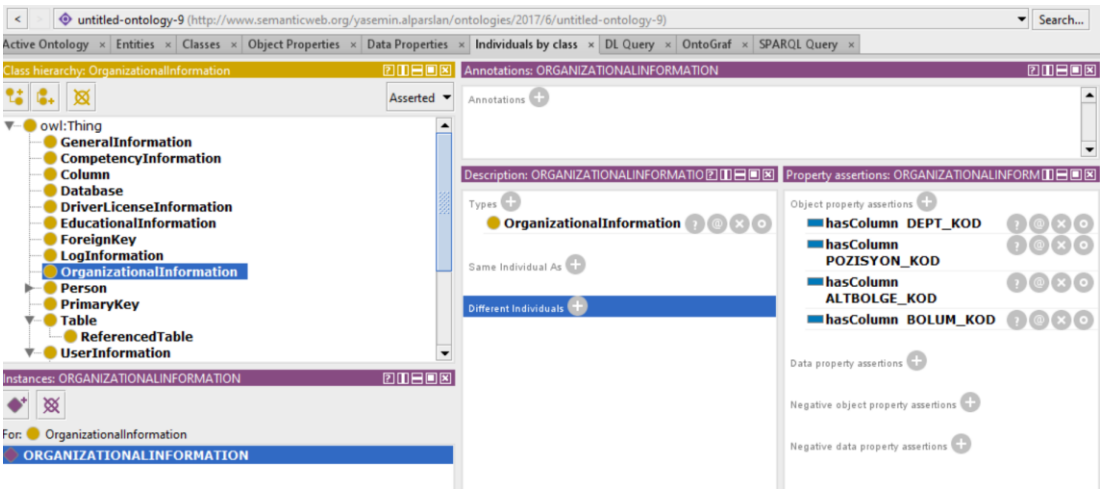


Figure 6. Organizational Information class in Protégé

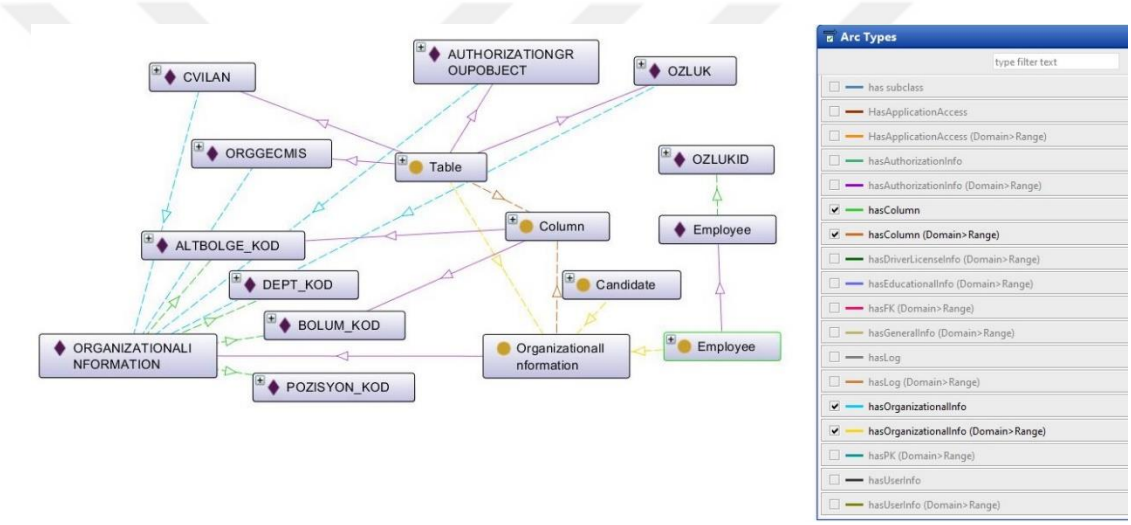


Figure 7. Ontology graph for OrganizationalInformation

- Some tables (ORGGECMIS, CVILAN, OZLUK) has organizational information.

For example, in real life according to the application, if a person has an OZLUKID, person is an employee. If a person has ADAYID, he/she is a candidate. An employee can have ADAYID, this means he/she started to company from recruitment process and was a candidate before.

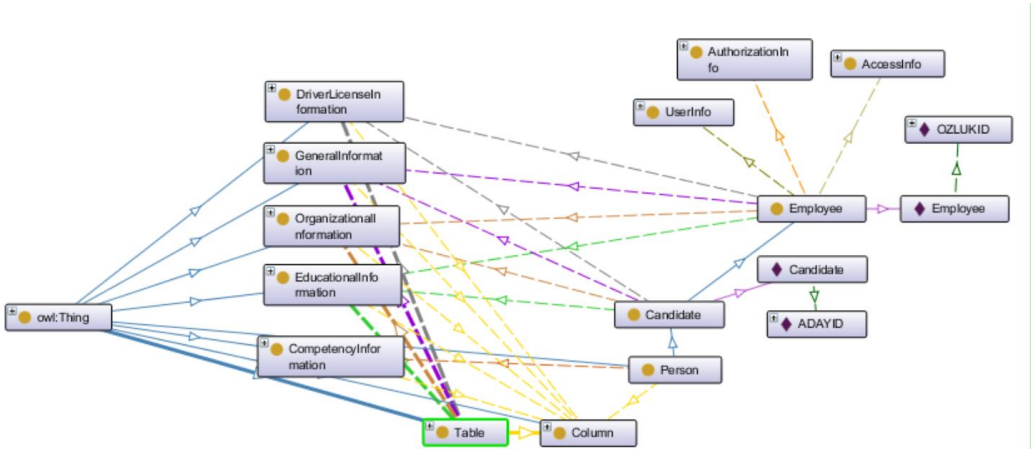


Figure 8. Relation between Person class and other classes with graph notation

An employee can have UserInfo, AccessInfo, AuthorizationInfo, DriverLicenseInformation, OrganizationalInformation, EducationalInformation, CompetencyInformation.

2.2.1.3 General Information

General information contains columns: SIGARA, ENGELLIBITTAR, ENGELLIBASTAR, HUKUMKATEGORI, ENGELLIACIKLAMA, SURELIENGELLI, HUKUMACIKLAMA, MEDENIHAL, SEYAHAT_EDEBILIR, CINSIYET, ENGELLIDERECE.

These columns are used in tables commonly: CV, CVILAN, OZLUK

Object property of this concept is hasGeneralInfo: A table(domain) hasGeneralInfo GeneralInformation(Range).

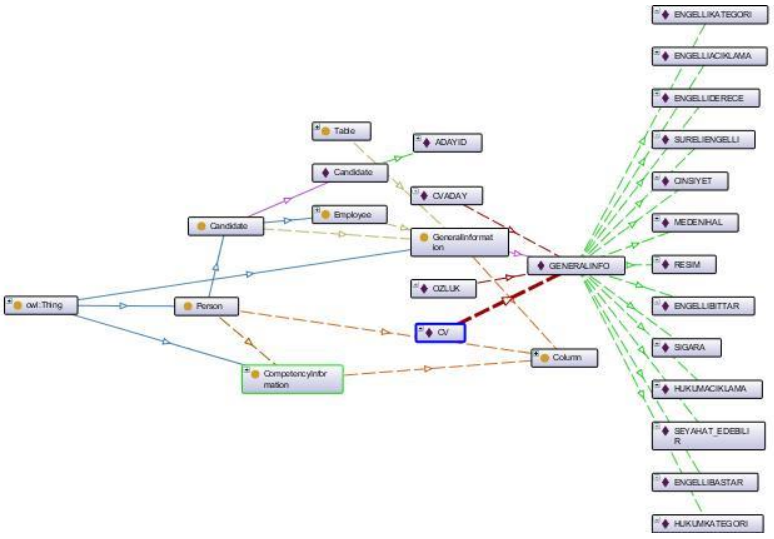


Figure 9. Ontology graph of general information concept

2.2.1.4 Roles in Database: UserInformation

Security of Database Management System can be considered in three steps

1. Authentication

2. Access Control

3. Audit

In this study, second one, "access control" is focused on with the hybrid model of RBAC and OBAC. A recent report by Alparslan, Komesli, Ünalır, Can (2015) mentions that users are assigned roles, and the roles are assigned access, users can have more than one roles in RBAC. RBAC restricts accessing to resource in accordance to users who belong to a group with the common aim or responsibility. After the assignment of roles to users, access of the source is provided by users who have separate roles. Every subject can do something according to its assigned role or roles. Subject can transact according to its role's rights [28].

According to this report, OBAC is defined as a mechanism for access control to resources with the system behavior using Ontology languages. OBAC provides access control to source with the semantic web based approach. Policies is formed source

based and parent data based together. Triples are the part of policies. Subject of triple is the one who access the resource. Thing is the source. Predicate is the policy objects. Policy objects that are also called deontic objects are composed of permission, prohibition, mandatory, special permission. OBAC provides judgement ability for decision mechanism and it allows listing, searching and finding access control information automatically.

UserInformation class in HR Ontology has 4 subclasses: AccessInfo, ApplicationInfo, AuthorizationInfo, UserInfo

AccessInfo can be considered as action such as insert, update, delete, list.

ApplicationInfo can be considered as the source that is wanted to be reached and used.

UserInfo can be considered as roles that access the source.

AuthorizationInfo can be considered as the limitation of the list on the screen of the HR application.

AccessInfo

AccessInfo has columns SORGU, KAYIT, DUZELTME, SILME. Employees can see the screens of HR applications and can do something on application according to their user rights which is equivalent to AccessInfo. The relation between access information class with other classes is shown in Figure 10.

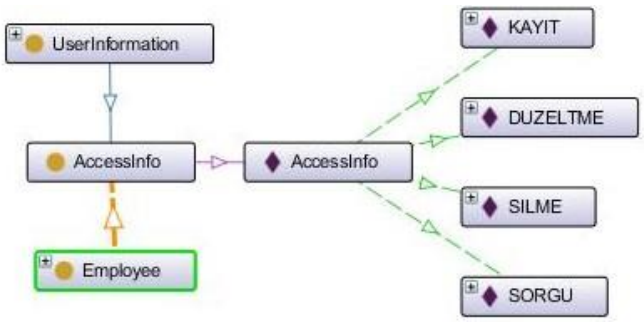


Figure 10. Access information of an employee

Column <SORGU> means listing right of employee. If an employee does not have SORGU right, then employee cannot see and list the rows in the application.

Column <KAYIT> means insert right of the employee. If an employee does not have the “KAYIT” right, then the employee cannot insert a new record in the application.

Column <DUZELTME> means update right of the employee. If an employee does not have the “DUZELTME” right, then the employee cannot update a record in the application.

Column <SILME> means delete right of the employee. If an employee does not have the “SILME” right, then the employee cannot delete a record in the application.

An employee can insert, update, delete, list a record for some modules and programs according to his/her AccessInfo rights.

ApplicationInfo

There is an HR application which contains modules and programs. ApplicationInfo can be considered as a resource. Users can have access to application. ApplicationInfo has sub classes: ModulInfo, ProgramInfo, ObjeInfo.

- *ModulInfo*

  ModulInfo has column MODUL_NO. HR Application has modules OrganizationalManagement, Employee, Absentee, Education, User etc. In this study, only modules in Table 3 are based. ModulInfo has programs. So, hasProgram object property is created.

  The relation between modules and programs can be expressed with “ModulInfo(Domain) hasProgramInfo (Object Property) ProgramInfo(Range)”. In Table 3 and Table 4, module names and program names are defined.

Table 3. Module names of HR applications

MODUL_NO	Modul Name
400	Definitions (Web)
401	Employee (Web)
404	User (Web)
407	Recruitment (Web)
411	OrganizationalManagement (Web)

- *ProgramInfo*

  ProgramInfo has column PROGRAM_NO. Column <PROGRAM_NO> refers to the screens of each modul. An employee can have access to a program

Table 4. Programs of modules in HR application and related tables

MODUL_NO	Modul Name	PROGRAM_NO	Program Name (Screen of Application)	Table Name in Database
400	Definitions (Web)	401	Section	BOLUM
		402	Department	DEPARTMAN
		404	Position	POZISYON
		405	Degree	KADEME
		410	Sub Region	ALTBOLGE
		411	Country	ULKE
		412	City	IL
		413	District	ILCE
		421	Faculty	FAKULTE
		480	Competency	NITELIK,YETKINLIK
		486	Position Profile Definition	POZISYONPROFILTASARIM
		487	Position Profile Matching	POZISYONPROFILSONUCLAR
401	Employee (Web)	4011	Employee Information	OZLUK
		4019	Employee Study	TAHSIL
		40111	Employee Organizational Information	ORGGECMIS
		40117	Employee Profile	PROFIL
		40121	Employee Driving License	EHLIYET
		4018	Employee Communication	ILETISIM
		40131	Employee List	OZLUK
404	User (Web)	4041	Create User	KULLANP
		4042	User Information	KULLANP
		4045	User Rights	KULLANHAKP,KULLANPROGP
		4044	Authorization Group	AUTHORIZATIONGROUP, AUTHORIZATIONGROUPOBJECT, AUTHORIZATIONGROUPUSER
407	Recruitment (Web)	4071	Candidate General	CV
		4072	Candidate Personal	CVADAY
		4073	Candidate Educational	CVTAHSIL
		4076	Job Advertisement	CVILAN, CVILANGRUP, CVILAN_IL,
		4077	Search In Candidate	CVADAY

In Figure 11, ApplicationInfo is related with employee and sub class of

UserInformation.

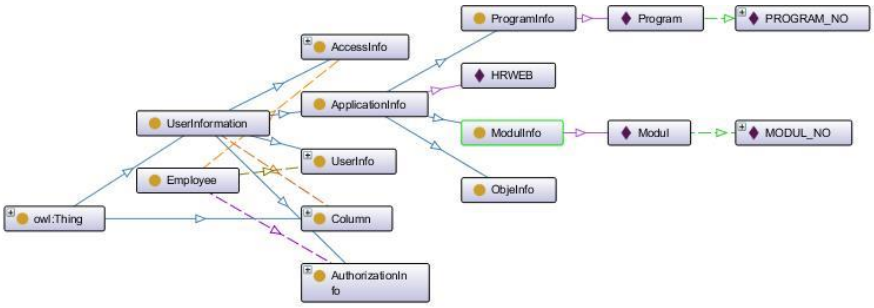


Figure 11. Application Information and its relations in ontology

- *AuthorizationInfo*

AuthorizationInfo sub class is created for the restricting the list when an employee wants to do something on screens after employee login the HR application. For example; if an employee's user information is assigned to a specific department name, employee will see and process on limited people which work on assigned department. In this condition, assigned department is called as object type. So, AuthorizationInfo provides to limit employee list according to user-object type assignment. In Table 5, standard object types of the HR database are shown. For example, If the object code=3, then a specific employee can see the limited employee list in modules according to the assignment of user- object type. An employee can have an AuthorizationInfo. In ontology, hasAuthorizationInfo object property is created.

An employee (Domain) hasAuthorizationInfo AuthorizationInfo(Range).

In Table 5, object types in Relational database is listed.

Table 5. Object types in DBMS

Related Table Name in DB: OBJETIPI	
OBJETIPKODU	**OBJEACIKLAMASI**
1	Cv
2	Aday
3	Özlük
5	Isletme
6	Görüsme
8	Alt Bölge
9	Bölüm
10	Departman
12	Pozisyon
15	Ilan

- *UserInfo*

It is observed that, all user information of the related tables has two columns: ISYERI, KULLANICI_NO. Column <KULLANICI_NO> can be considered as userid and column <ISYERI> can be considered as the company that employee works on. All user primary key is made up by these two columns. The sense is that an employee can have same userid for different companies. In other words, an employee can have an access for more than one company with the same userid. Hence, userid is not enough for the user information. Userid and company name together provides unique singularity. The relation is shown in Figure 12.

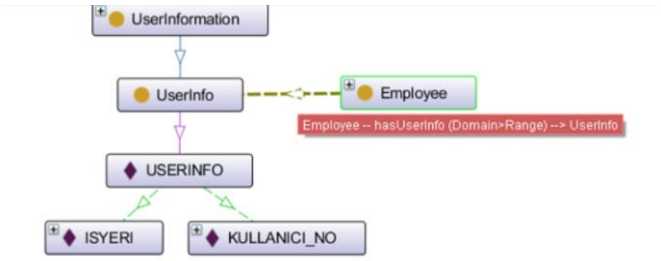
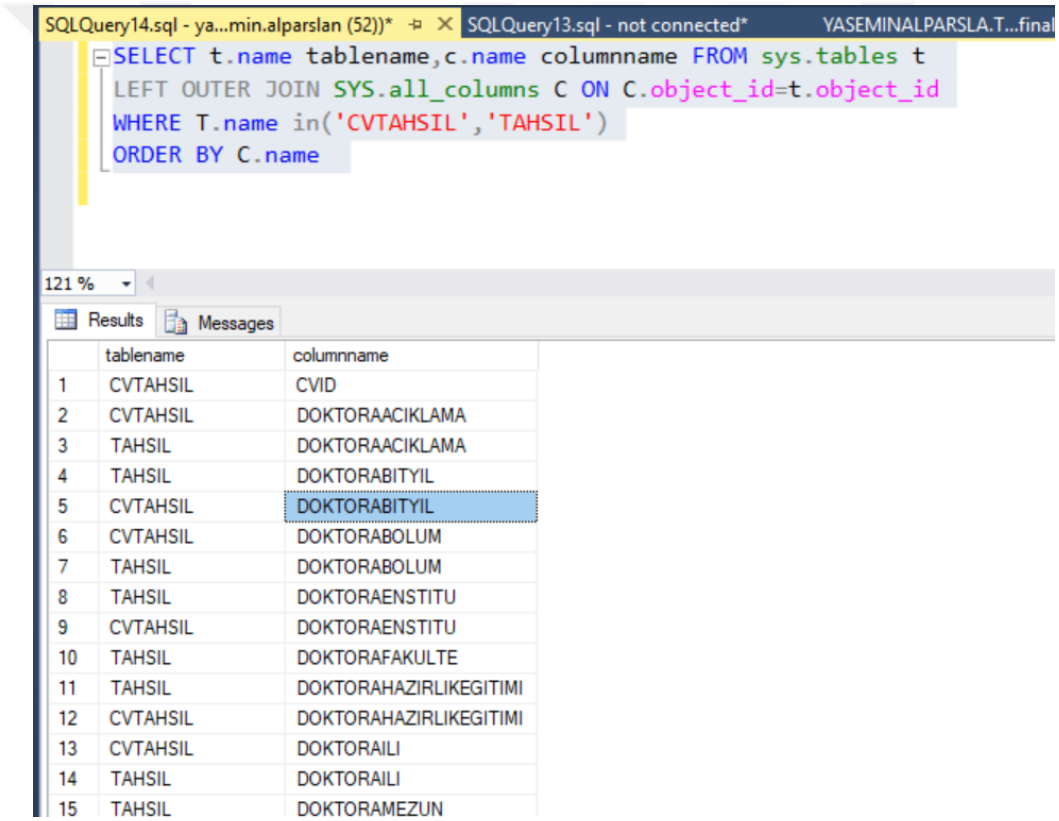


Figure 12. Relation between user information and employee

2.2.1.5 Educational Information

A CV of a candidate can have educational information. And employee can have also educational information with same column names. According to usage of the database, there is a person class. A person is a thing and has 2 subclasses as “candidate” and “employee”. All of the rows of OZLUK table are each can be considered as an employee. All of the records of CVADAY table are considered as candidate. When we look at the Figure 13, simplification is needed for the repetition of the columns for different tables..

As we mentioned before, benefit of the ontology is preventing mess of information.



	tablename	columnname
1	CVTAHSIL	CVID
2	CVTAHSIL	DOKTORAACIKLAMA
3	TAHSIL	DOKTORAACIKLAMA
4	TAHSIL	DOKTORABITYIL
5	CVTAHSIL	DOKTORABITYIL
6	CVTAHSIL	DOKTORABOLUM
7	TAHSIL	DOKTORABOLUM
8	TAHSIL	DOKTORAENSTITU
9	CVTAHSIL	DOKTORAENSTITU
10	TAHSIL	DOKTORAFAKULTE
11	TAHSIL	DOKTORAHAZIRLIKEGITIMI
12	CVTAHSIL	DOKTORAHAZIRLIKEGITIMI
13	CVTAHSIL	DOKTORAILI
14	TAHSIL	DOKTORAILI
15	TAHSIL	DOKTORAMEZUN

Figure 13. Educational Information for both candidate and employee

“EducationalInfo” class is created.“hasEducationalinfo” object property is created. A table(Domain) “hasEducationalInfo”(object property) EducationalInfo (range). Ontological relation of educational informatşon is shown in Figure 14.

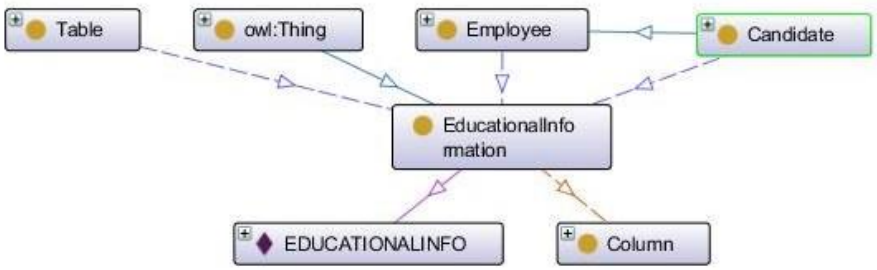


Figure 14. Educational Information relations

An employee and candidate has educational information. A table has educational information. And educational information has many columns.

2.2.1.6 Driver Licence Information

CVEHLIYET table takes driver license information of candidates.EHLIYET table takes same information for employees.

```
SELECT t.name tablename,c.name columnname FROM sys.tables t
LEFT OUTER JOIN SYS.all_columns C ON C.object_id=t.object_id
WHERE T.name in('CVEHLIYET','EHLIYET')
ORDER BY C.name
```

121 %

Results | Messages

	tablename	columnname
1	CVEHLIYET	BELGENO
2	EHLIYET	BELGENO
3	CVEHLIYET	CVID
4	CVEHLIYET	DUZELTMETARIHI
5	EHLIYET	DUZELTMETARIHI
6	EHLIYET	DUZELTMEYAPANKUL
7	CVEHLIYET	DUZELTMEYAPANKUL
8	CVEHLIYET	EHLIYETTIP
9	EHLIYET	EHLIYETTIP
10	EHLIYET	ID
11	CVEHLIYET	KAYITTARIHI
12	EHLIYET	KAYITTARIHI
13	EHLIYET	KAYITYAPANKUL
14	CVEHLIYET	KAYITYAPANKUL
15	EHLIYET	OZLUKID

Figure 15. Common columns names for driving license information

These two table has same columns. Common columns are clustered in

DriverLicenseInformation as it is shown in Figure 16.

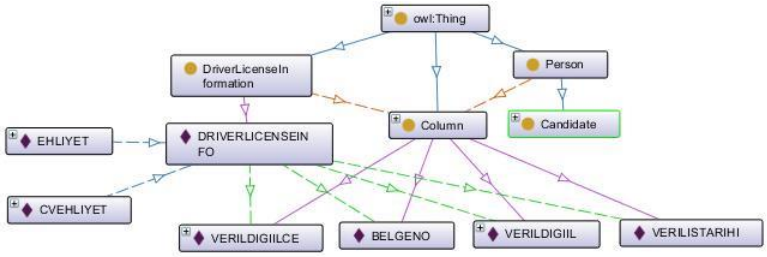


Figure 16. Driver license relation in ontology

2.2.1.7 Competency Information

A position requires some competencies. For example; a software engineer should have known JAVA in advanced level in a company. But an employee who works on this position knows JAVA in beginner level. So, employee doesn't satisfy his/her position's requirement. Comparison and matching of position's competency and employee's competency is a critical point for HR responsibilities. If this condition is occurred then HR should plan an education plan for the employee's. Also, career path of this employee will depend on his development efforts and educations. By the reason of this efficiency bonus would be given or not that employee or would be discharged for a just cause. Also in recruitment process, finding appropriate candidate according to comparison of candidate's competency and position of the job advertisement provides happy and efficient employees in company.

In Figure 17, the relation of competency is shown. An employee can have a position. A position is a thing and a position can have CompetencyInformation. CompetencyInformation has some columns such as YETKINLIKID etc. A candidate can have a competency information also.

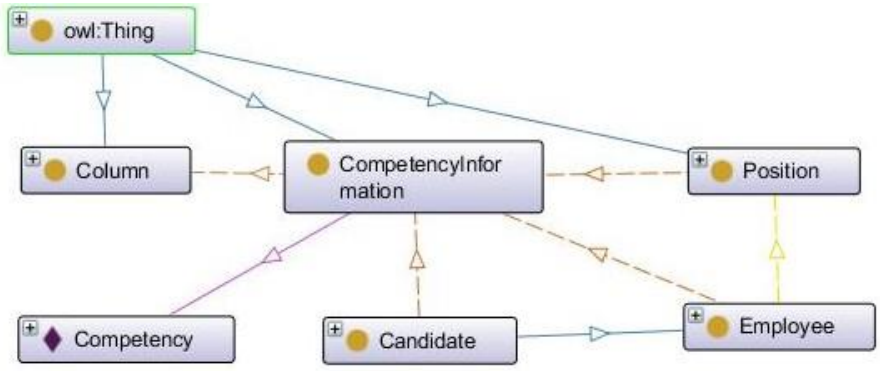


Figure 17. Competency relation between position and position in ontology graph

2.3 Data Representation of RDB To Ontology

LinkedDataTools.JenaDotNet tool is used to create instance of the classes with C# programming language in Microsoft Visual Studio 2015 version. Metadata and data transformation from relational database to OWL is provided. Metadata transformation is made from SQL Server system objects. SQL server used version is SQL Server 2016.

HR database has tables, columns, primary keys, foreign keys. Before the developing a method and an implementation tool for the access control framework, there should be a mapping model between RDB schemas and data itself in the system objects.

Mapping RDB to OWL is considered many times up until now in different domains. In one of the similar study, mapping is provided in Java using Apache Jena tool for mapping columns to tables in ontology [29].

Database diagram of the database can be seen in Appendix 1. Table names are listed in Appendix 2. Column names are listed in Appendix 3. This study is more than relational owl. Because, conceptual grouping between database objects and HR ontology is taken in consideration when modeling ontology.

In one of the similar study, there were also proposed an approach for ontology mapping ffrom RDB to OWL with the background knowledge by using a concept hierarchy for managing knowledge [30].

RDB2OWL mapping is presented again in this study considering database-to-ontology relation for managing knowledge. Relational database semantic reengineering is

involved for representing a human readable RDB-to-RDF/OWL mapping. For example, ObjectProperty maps the relation between tables, relevance of columns to tables [31].

For example, in [32], a record in a table in RDB refers to a RDF Node. RDF predicate refers to column name of RDB Table. This approach also can be called as “Domain Ontology Mapping”.

Figure 18. Mapping representation between ontology and RDB

In Figure 18, the transformation between ontology and RDB is shown.

Syntax Presentation

Mapping definition between RDB and ontology is HR domain is emphasized in this part of the study.

For example, a database diagram which is related with the position and position’s competency level is shown in Figure 19.

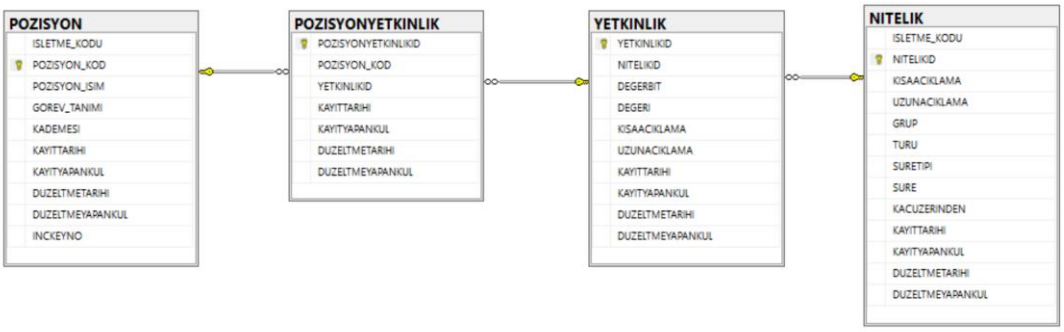


Figure 19. An example for tables and columns in proposed model

According to Figure 19, some table expressions are shown in a syntax below;

- *POZISYON*
- *POZISYON P, POZISYONYETKINLIK PY;*
  P.POZISYON_KOD_FK=PY.POZISYON_KOD
- (POZISYON P, POZISYONYETKINLIK PY;
  P.POZISYON_KOD_FK=PY.POZISYON_KOD) PYF

Object property mapping is defined with some references. Source class reference is shown <s> and target class reference is shown with <t>.
Reference between classes in ontology is shown with <s> and <t> marks.

Object Property expressions are shown below with these defined references:
- <s>, <t>; <s>.POZISYON_KOD_FK = <t>.POZISYON_KOD
- (Position {uri=('Position',POZISYON_KOD)}) <s>,
(Competency {uri=('Competency', POZISYONYETKINLIKD)}) <t>;
<s>.POZISYON_KOD_FK = <t>.POZISYON_KOD

CHAPTER 3
CONVERTING RBAC TO OBAC

3.1 Policies for Access Control Mechanisms in HR Case Study

Policies are the concept of a behavior. Rules defines the behaviors. And policies can be considered as a set of rules. Sloman [33] presents a policy perception in which he relates subjects and targets with each other. He gives a special interest in describing and comprehending policy. Two seperate groups appears as it being categorised by him in the names of authorization and obligation. He determines specific boundaries named temporal and parameter value regarding to policies. [33]
Policies shown with the triples. Triple objects are: event, condition and action.
Policies are applied to domains. Objects that take place within the domains are in a specific behaviour and policies have influence on them directly and indirectly [34].

A rule example;

"HR specialist can record a new employee to the HR application If a Human Resources Specialist has competency "Legal, Government and jurisprudence"

HasUserRight {x, AccessInfo (insert, employee (x, (hasCompetency (y,CompetencyInformation(has, "Legal, Government and jurisprudence"))}

A condition of an action can be changed so policy can be dynamic [35].

3.1.1 Policy Definition Terms

There are some definitions about policies.

Policy: A set of rules.
Authentication: Verification of user authorization to source.
Access control: Being auditable to access of information is can be called as access control.
Role: Special authorization group that is given the users.
Permission / Right: Authorization granted to someone who access the source or

information.

Prohibition: Authorization not granted to someone who access the source or information.

Authorization: Actions on the data that are permitted.

Speech Act: Provides the security control and to be changed policies dynamically.

Policy Engine: It does inference and annotates by using speech acts and domain. [35].

3.1.2 Policy Examples

There are some policy examples for HR domain.

- Only employees who work in Human Resources Department can insert a new record of an employee general information. (Permission)
    - Has(x,right(AccessInfo.INSERT,Employee(x,'HumanResourcesDepartment')))
- Only employees who work in Human Resources Department can update a new record of an employee general information. (Permission)
    - Has(x,right(AccessInfo.UPDATE,Employee(x,'HumanResourcesDepartment')))
- Only Human Resources Specialist can assign an employee to a position. (Permission)
- An employee can not be assigned a position if he doesn't correspond that position's competency level. (Prohibition)
- If an employee doesn't have driving license information, then employee can not be assigned to Project Specialist Position. (Prohibition)

3.2 Automatic conversion of RBAC to OBAC

In literature, there are similar studies. For example, one of the study presents SO-RBAC (Semantic Ontological Role-Based Access Control). The traditional RBAC is translated to proposed ontology-based RBAC models by using Prolog and OWL-DL language with using SWRL for reasoning. In this study, Prolog is used to manage rules into OWL/SWRL with OWL-DL and OWL- Full languages which are better for reasonability

[36].

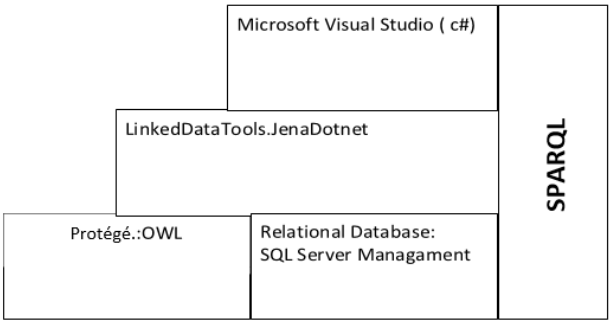


Figure 20. Used tools for the architecture

OBAC model is superior that RBAC because RBAC policy things have only permission and prohibition but OBAC model has also mandatory and special permission except only permission, prohibition [35].

3.3 Policy Languages

There are many policy languages to provide security for semantic web. Some policy languages are denoted as following sub titles.

Semantic web policy language requires:

- Ability to make context for agents proceeding and defining policies at a meta abstractive way.
- Ability to define actions as they are being permitted or forbidden according to sources (authorization or limitation can be determined)
- Ability to describe actions for resources like obligation

3.3.1 Rei

Security is a critical issue for semantic web as well as the other technologies. Declarative policies are flexible and simple against to the challenges of heterogeneous entities. Rei is a policy language in OWL-Lite. It is occurred by conflict decision, speech acts, logic for managing variables and specific relation mappings. Rei provides to manage permissions,

obligations of an entity in OWL language [37]. KAOS get advantage of interpreting domain or policy ontologically and conflicts statically [38]. Permissions, prohibitions, obligations and special permissions of users can be defined by Rei policy language. In specific domain's ontology, policies can be defined in their ontologies by using Rei [39]. Rei concepts are mentioned in Table 6.

Table 6. Rei Concepts [35]

Term of Rei	Definitions
Policy	A rule set defining the behaviours of the elements within its orbit
Policy domain	Elements and sources effected by policy
Deontic object	Permission, prohibition,obligation,dispensation
Speech act	Using of language to realize specific acts
Meta policy	Policy determining how another policy would be used. It refers to mechanisms used to solve contradictories in policy language
Constraint	A condition related to a policy,rule or deontic object

When we take Rei's perspective it is seen that Kagal [34] concerned authorization, obligations, prohibitions and dispensation rules contributed to policy. His separation occurs under three groups called actions, constraints and objects of policy. He takes these boundaries as centralised upon domain and from one point externally oriented from policy specification point [34].

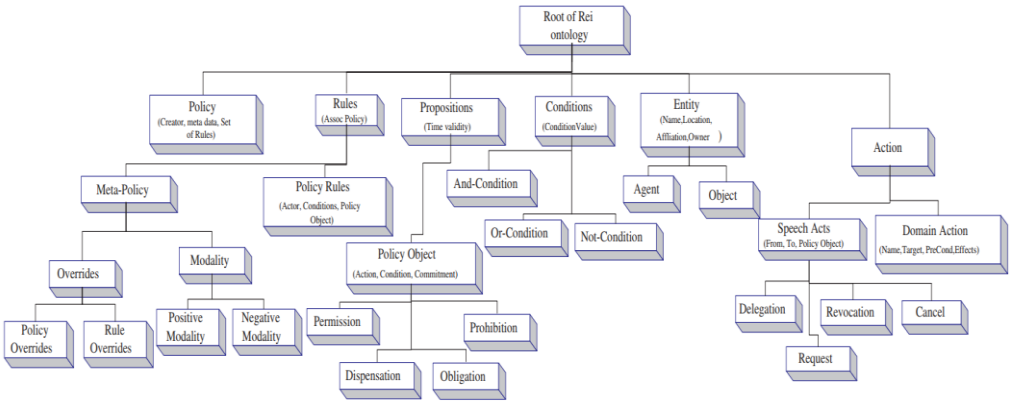


Figure 21. Rei policy language ontology [40]

Rei policy language has meta policy definitions for solving conflicts. Meta policies determine how conflicts are comprehended and solved dynamicly.

Policy shows conflicts in two ways: Conflict between policies or between rules are solved by determining priorities and superiorities. Superiorities signifies rules related to solution of conflicts and levels between policies. For example, A1 rule may write over B1 or A1 policy write over B1 policy [35].

Obligation and Prohibition conflict may occur a necessary act's being come across with ban on a set of targets. But Rei obligates permission before performing an act and reduces the problem.

3.3.2 KAoS

OWL language allows description when it comes to KAOS policy. It brings policy and effect sphere management with policy for pattern presentations. KPO describes a KAOS policy ontology. Authorizations refer to constraints which allow or restrain act and obligations which refer to necessity dependent acts or bans for giving up on them). KPO contains both [39].

KAoS presents control services for policy domain for agent or distributed computing platforms. KAOS has authorization and obligation. KAOS gives policy limiting agents allowed or obliged acts. Each policy has a control task. Action groups draws subject target or context. Property limits occur.

Policy can be evaluated with actor's class that indicates property. KAOS gives categories according to property limitations and that may draw a policy regarding to context conditions. Two basic properties, i.e., the hasDataContext and the hasObjectContext properties, and their subproperties are the elements of act context. KAOS ontology's sub properties are performedBy (actor), accessedEntity (period and the target source) etc.

Examples presents KAOS as ontological approached policy viewer which exploits OWL. KAOS get advantage of interpreting domain or policy ontologically and conflicts statically. [38].

KAOS and Rei contradict with each other at the point of allowing a retrieving policy to control. Rei doesn't approve this kind of policy disclosure. User should observe the process by accesing to see if it can or not demanding a printer. Rei has disadvantage about problem or contradict solving just because of this feature. KAOS become advantegous comparing to it. It is specific though [38].

3.3.3 XACML

XACML is a policy model which is used by many Access Control Systems. Fundamentally it contains a syntatic context within a language and based on semantically proceeding what is foreseen. Demand and its reponse are the subject of inquiry. It uses a standard model as an interfce. That takes place through PDP (Policy Decision Point) and it provies a general behaviour. PEP (Policy Enforcement Point) handles demands and requirements. A PEP is localised within a system as it can be in Figure 1. IETF derived policy concepts are the source [41]. There should be a usage guide and perception for eXtensible Access Control Markup Language (XACML). "Core" and "hierarchical" role based access control (RBAC) (pointed out in [ANSI-RBAC] have basic demands and to provide those will be thanks to it.

XACML has elements differantiated like an attribute name identifier, a data type identifier, and an attribute value which are in relation with subjects (as can be called Subject Attribute), resource protection (Resource Attribute) and Action Attribute which defines act realized on resource or the environment of the Request (Environment Attribute).

XACML subjects are applied for users. XACML SubjectCategory values are for usage as their being appropriate. XACML Subject Attributes are defined by roles. According to policy sphere and application we may get sense of role groups [17].

3.3.4 Ponder

Ponder is written by authors who brought some establishingand regulating rules in terms of policies. These include a choice which shows itself within the system's proceeding. Mechanism becomes differantied from policy. It provides a change in systems proceed. Implementation remains stil but behaviour gets affected and changed via the segment.
In the character of Ponder language we observe an object based and expressive side while handling issues like management or security. Thanks to this feature, policy types can be viewed diversely by which they can be passed to make a determined instance. It functions so that is used by Rei spontaneously. He categorizes them as actions and objects of policy. He gives them opportunity to be tied upon each other vigorously [23]. Ponders model consist of authorization, obligation, refrain and delegation based basic policies and roles, relations, control based composite policies.

3.3.5 WSPL

WSPL is Web Services Policy Language and gives ability to categorize many policies. Some are authorative concepts and some are related to act, quality, security. By dividing policies, we may have both sided satisfaction in a single policy. Dependency appears related to time or cost kind of attributions. Standard model for a policy parameter provides a standard engine which can be suitable for policies [42].
OASIS eXtensible Access Control Markup Language (XACML) Standard [2 - S. Godik, T. Moses, eds., "OASIS eXtensible Access Control Markup Language (XACML) Version 1.1" comprises of WSPL's syntax [43].
WSPL policy contains rules related to choices applicable. Rules order is determined by various factors like preference. Predicate part forms a WSPL rule. Predicates may limit. (The limited users are categorized as equals, greater than, greater than or equal to, less than, less than or equal to, set-equals, and subset.) in "A > 3" a predicate's limitation level is viewed or as in "A > B" a predicate's limit over another attributed is viewed). All the predicates in a rule must be satisfied for the rule to be satisfied. Web service target appears in policy [42].

3.4 Case Study in HR

In this study, user rights for employee who works in HR department and other departments and candidates are considered as a role.

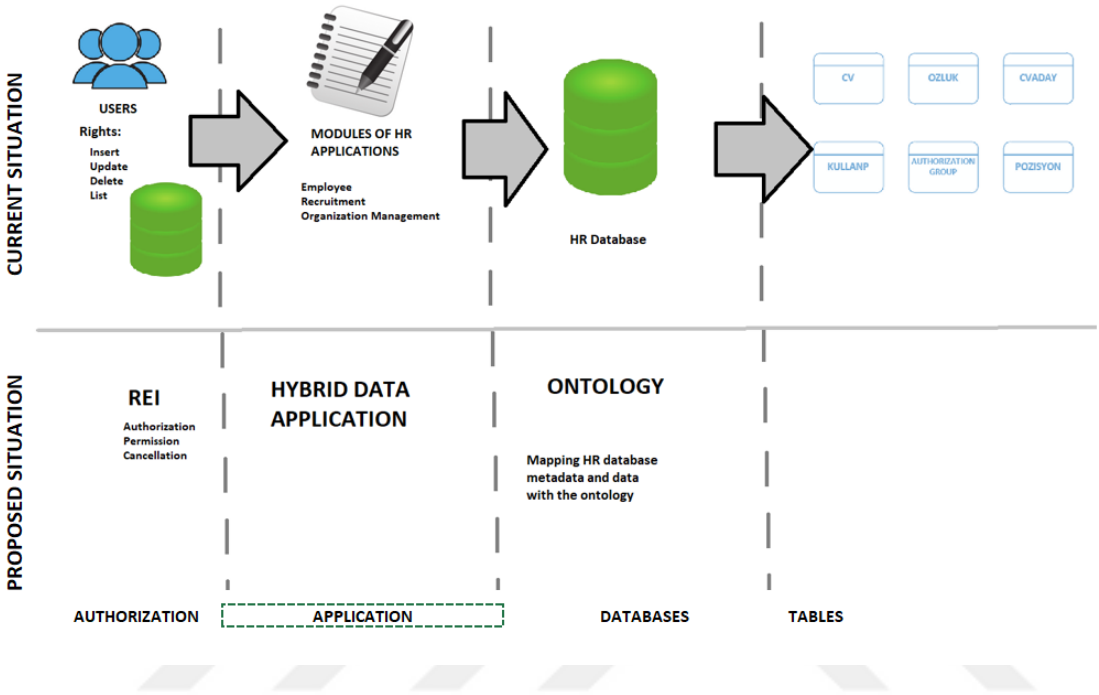


Figure 22. HAC structure for HR case Study [28]

Scenario of the case study

XYZ Company has 400 employees. Human Resources Department is occurred by one HR Director, one HR Manager, two HR Recruitment Specialist, one HR Education Specialist, three HR Payroll Specialist. Merve is HR Manager. Duygu and Kemal are HR Recruitment Specialist. Melis is HR Education Specialist. Mustafa, Yeliz and Cem are HR Payroll Specialist. Mehmet is a IT Specialist. In summer, HR department has two trainees.

Trainees can not record a new employee. They can update employee's general and educational information. Company gives considerable importance to employee's development and company plans educations in accordance to their career path. Melis can insert, update and delete only educational information of employees.

Duygu and Kemal are responsible for the resume pool of the company. They can list the candidates, but they can not delete the CV's of any candidate. Only HR Recruitment specialist can take on company.

Only Murat, Yeliz and Cem can enter the new employee. If an employee has Netsis Payroll module knowledge, then employee can enter the new employee record. (Even a trainee can record a new employee). If a HR specialist has "Legal, Government and jurisprudence" competency then HR specialist can record a new employee to the HR application by using Employee Module and Employee Information screen.

After the new entrance to organization, after two months, no one can change the identity number of any employee because it is unique and not updateable.

If an Mehmet has SQL knowledge, then he can manage HR SQL database.

3.4.1 Policies of Case Study

Permission: IT Specialist can list HR SQL Database in accordance to his being SQL knowledge.

hasAccessInfo (x, right(ACCESSINFO.SORGU,Employee((x,hasPosition(ITSpecialist), 'SQLKnowledge')))

Permission: If an employee has Netsis Payroll module knowledge, then employee can enter the new employee record.

Prohibition: Trainees can not record a new employee.

hasAccessInfo (x, dispensation (x, (Employee(hasPosition(x,'Trainee'))

Prohibition: After the new entrance to organization, after two months, no one can change the identity number of any employee because it is unique and not updateable.

Only employees who work in Human Resources Department can insert a new record of an employee general information. (Permission)

Has (x, right (AccessInfo.INSERT,
Employee(x,'HumanResourcesDepartment')))

Only employees who work in Human Resources Department can update a new record of an employee general information. (Permission)

Has (x, right (AccessInfo.UPDATE,
Employee(x,'HumanResourcesDepartment')))

3.4.2 Policy Ontology Design

In the first part of the study, an ontology mapping between HR relational database and HR ontology is made with a different approach. We have a HR ontology and we want to make an access control. Because of this there is needed to get a way of providing ontology based access control. There are many policy ontologies. Rei is used as policy language in this study.

HR Policy Ontology is designed based on HR ontology that is created for this study.

Steps for the create a policy ontology are:

- Namespace definition of Rei policy ontologies
- Entity: Variable class creation: In this step; actors who are used for the rules are created.
- Constrain: SimpleConstraint class creation: In this step; boundries are defined. Such as constraint: subject, constraint: predicate and constraint: object triple objects.
- Action: DomainAction class creation: In this step actions that are used in rules are defined. Action: Actor means the actor who will do in rule, action: Location means location of action. For defining the constraints constraint: SimpleConstraint or more complex contstraint definition action: precondition should be defined if necessary.
- According to policy object; deontic: Permission, deontic: Prohibition, deontic: Obligation or deontic: Dispensation should be defined.
- End of the all steps, policy: Policy class is created. Examples under this class should include policy: actor, policy: action, policy:grants properties.

HR Policy Ontology design includes: HybridBasedAccess_HR_Ontology, ReiAction, ReiDeontic, ReiEntity, ReiPolicy ontologies. In Protégé, all ontologies are merged as in the Figure 23. With this process, Rei ontologies and HR ontology came together and

creates and HR Policy Ontology.

Rei ontologies can be reached from https://www.csee.umbc.edu/~lkagal1/rei/ontologies/ .

Included ontologies and their contents are told below:

HybridBasedAccess_HR_Ontology: It is more from a relational owl, also with a case study based approach, relational HR database is transformed into a specific HR ontology. There are many approaches to mapping relational data to RDF. 3 of them are mentioned below:

Direct Mapping: Supplying a relational database (schema and data) plus a stem URI defines an RDF graph, which emulates the relational schema.

Direct mapping Plus Ontology Mapping: The RDB2RDF may define a mapping semantics as a mapping to a direct graph, followed by the application of RDF graph transformations into an RDF graph in a final ontology.

Database to Ontology Mapping: The RDB2RDF may define a mapping semantics as a single step process from database to an RDF graph in a final ontology.

ReiAction: Ontology includes properties that are required for all actions.

ReiDeontic: Ontology includes deontic objects such as dispensation, permission, prohibition. It includes the conditions between actor and action.

ReiEntity: Ontology includes rei entities such as agent, object, variable.

ReiPolicy: Ontology defines the behaviours of entities in a specific policy domain. It includes rules and policy contexts.

ReiConstraint: Condition is used for defining entity or entity subset and action subset. There are two types of condition: SimpleConstraint and BooleanConstraint. SimpleConstraint, consist of triples like RDF: subject, predicate and object. BooleanConstraint sonsist of both simple constraint and Boolean conditions like “And”, “Or”,” Not”.

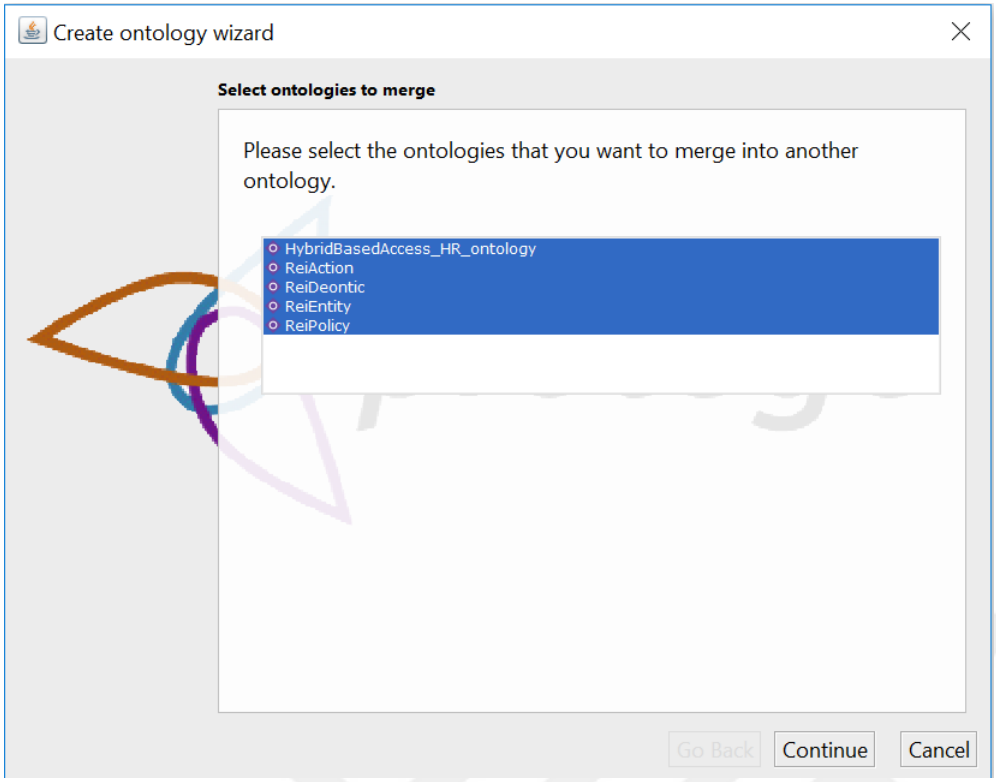


Figure 23 HR Policy Ontology Design

All HR policy ontology classes are shown in Figure 24.

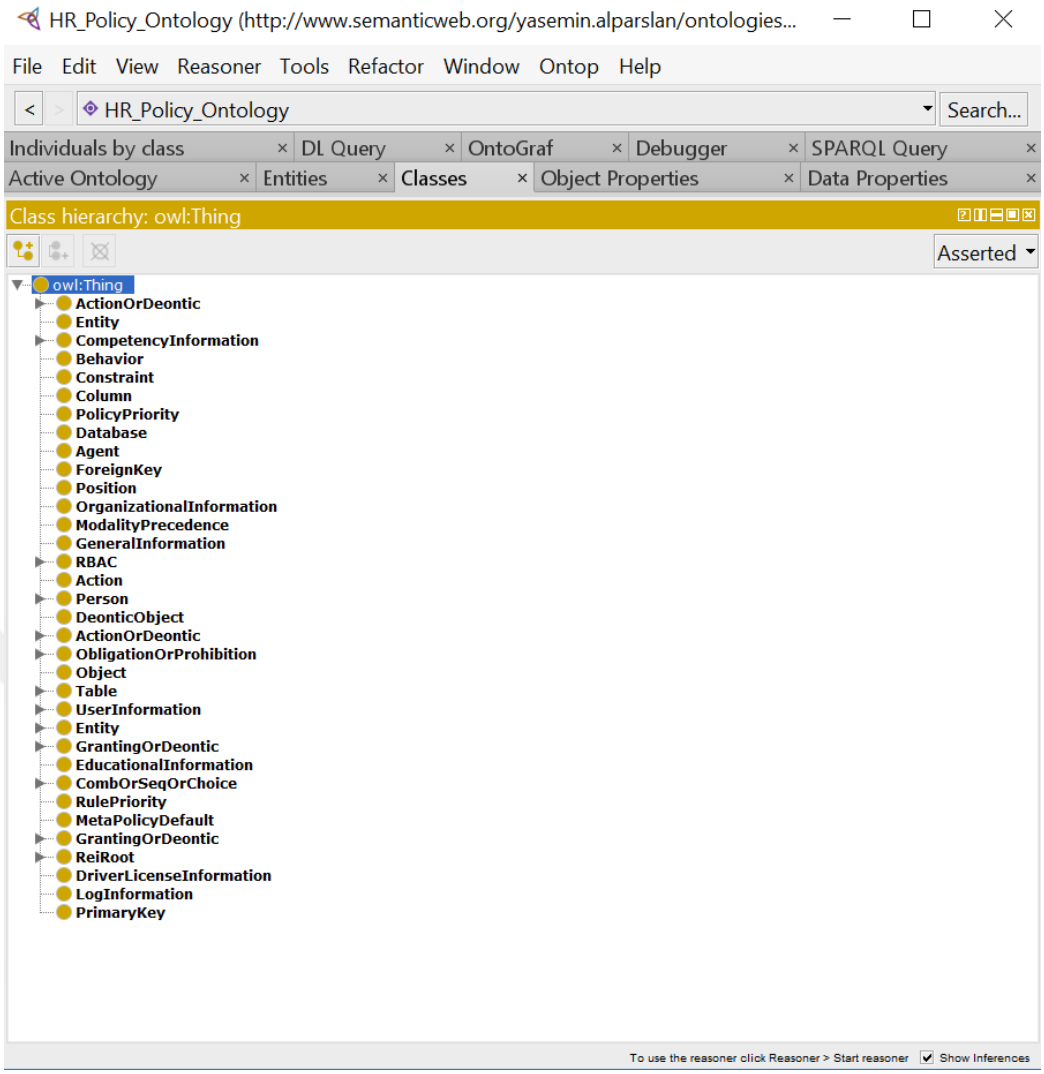


Figure 24. Classes of HR Policy Ontology

In HR Policy, a permission is defined as "If an IT Specialist has SQL Knowledge, then he can manage HR SQL database." Access permission is provided by DeonticActor, DeonticAction and DeonticConstraint.

Deontic Actor is the one who is related with the action. For this permission, DeonticActor is provided from ontology IRI that is created for this study specifically "http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/6/HybridBasedAccess_HR_ontology". Ontology IRI has positions of the company knowledge.

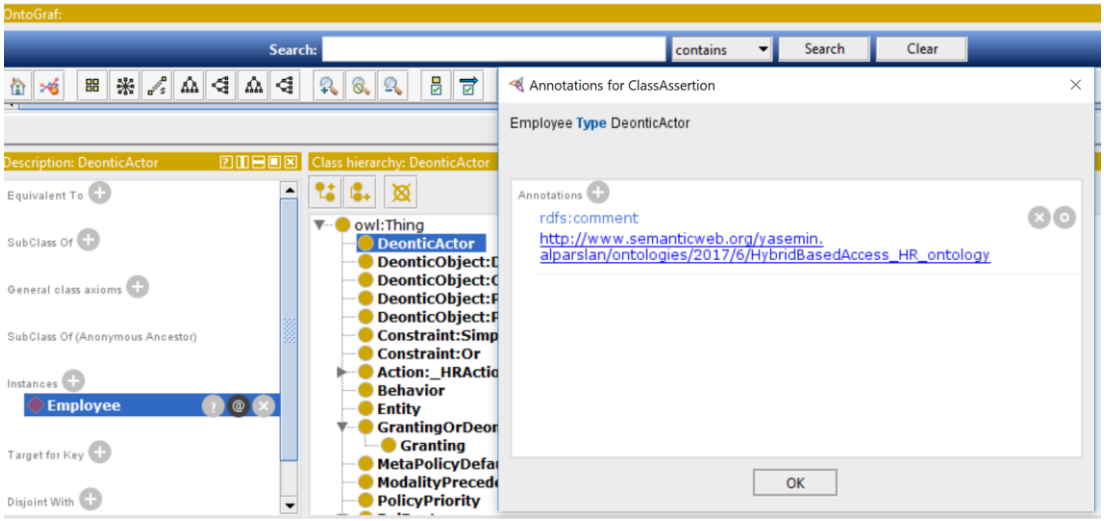


Figure 25. Deontic actor of the policy ontology

According to the scenario, HR Recruitment Specialist, HR Payroll Specialist, Trainee, IT Specialist are actors of the HR Policy ontology. In other meaning they are entities.

- Rei Simple Constraint example below

```
<!-- constraint describing HR Specialist members -->

<constraint:SimpleConstraint rdf:ID="IsHRSpecialist">

        <constraint:subject rdf:resource="#var1"/>

        <constraint:predicate rdf:resource="&rdf;type"/>

        <constraint:object
rdf:resource="&Employee;OrganizationalInfo:Position"/>

        <policy:desc>HRPolicy</policy:desc>

</constraint:SimpleConstraint>
```

- Rei Permission Example:

```
<!-- specific Rei permission -->

<!-- Rei permission giving HR Specialist the permission to add new employee in Employee Module -->

<!-- but this permission is not linked to the policy, so its not active -->

<deontic:Permission rdf:ID="Permission_AddEmployee">

        <deontic:actor rdf:resource="&inst;Employee"/>

        <deontic:action rdf:resource="&inst;AddEmployee"/>

        <deontic:location rdf:resource="&inst;ApplicationInfo:ModulInfo"/>

        <deontic:startingConstraint rdf:resource="#hasPosition"/>

        <deontic:endingConstraint rdf:resource="#hasCompetency"/>

</deontic:Permission>
```

Rei Obligation Example:

```
<!-- Generic Rei Obligation  -->

<!-- Rei obligation trainees update employee identity information -->

<deontic:Obligation rdf:ID="Obligation_UpdateInfoforTrainees"

      policy:desc="Obligation to update employee information when employee is trainee">

      <deontic:actor rdf:resource="#employee:trainee"/>

      <deontic:action rdf:resource="&inst;ModificationEmployeeInfo"/>
```

Policy of the university example:

```
<!-- policy of CS Department -->

<policy:Policy rdf:ID="CSPolicy">

        <policy:actor rdf:resource="#var1"/>

        <policy:context rdf:resource ="#IsMemberOfCS"/>

        <!-- granting instance with no additional constraint -->

        <policy:grants rdf:resource="#Perm_StudentCSPrinting"/>

        <policy:grants rdf:resource="#Proh_StudentCSPrinting"/>

        <policy:grants rdf:resource="#Perm_BobJones"/>

        <policy:grants rdf:resource="#Perm_TimDelegateFacultyCSPrinting"/>

        <!-- granting instance with additional constraints -->

        <policy:grants rdf:resource="#Granting_FacultyPrinting"/>

        <!-- granting instance with no additional constraints -->

        <policy:grants rdf:resource="#Obl_GradStudentRegister"/>

</policy:Policy>
```

- Any human user, software agent or hardware resource can be defined as an entity: Entity. Currently, the Entity class has only one property called as entity: affiliation, which is used to specify what organization an entity belongs to.
- entity: Agent, entity: Object and entity: Variable are the subclass of entity class.

Entity can be an agent (human or software), a hardware resource or a variable that is used to describe constraints.

In this study, entity includes Person, Module and ProgramInformations. Person: Candidate and Person: Employee are subclass of the Person class; so subclasses are also entity.

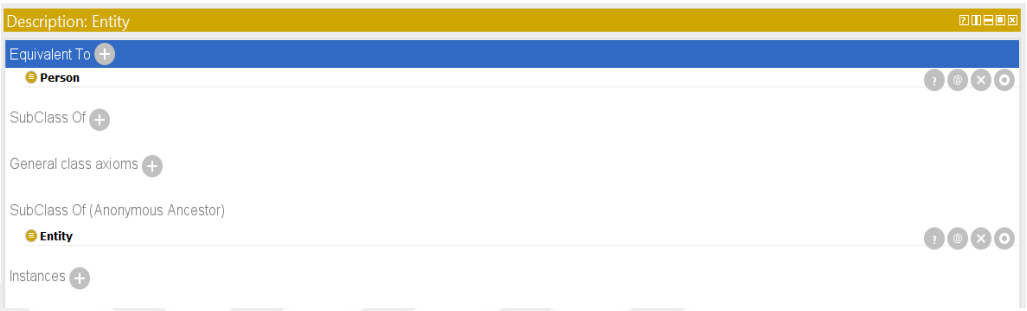


Figure 26. Entity class of Rei equivalent to Person class

HybridBasedAccess_HR_Ontology has AccessInfo subclass under UserInformation class. AccessInfo includes insert, update, delete and list rights for modules and programs. AccessInfo property assertions are matching to DomainAction as in Figure 27.

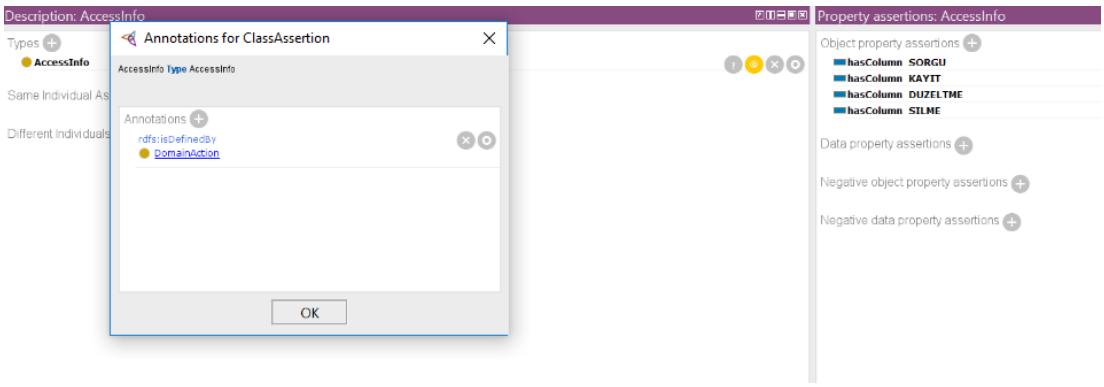


Figure 27. Domain Action in HR Policy Ontology

```
<ClassAssertion>
    <Annotation>
        <AnnotationProperty abbreviatedIRI="rdfs:isDefinedBy"/>
<IRI>file:/C:/YASEMIN/YASAR%20EDU/____GRADUATE%20THESIS____/___
_Thesis____/c%23/OntologyModelling/Semiodesk_TrinityModelling_Ontology/Ont
ologyModelling/Ontologies/ReiAction.owl#DomainAction</IRI>
    </Annotation>
    <Class
IRI="http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/6/untitled-
ontology-9#AccessInfo"/>
    <NamedIndividual
IRI="http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/6/untitled-
ontology-9#AccessInfo"/>
  </ClassAssertion>
```

- DomainAction is subclass of action: Action. It consists of additional properties for describing application or domain specific actions. For example, in HR domain-specific actions include adding new employee, updating employee information, listing employee's information, deleting employee's informations, promoting employee.
- action: target property corresponds the object on which the action is performed. When we say, "an action that can be performed by anyone as long as the target is an access right to modules and adding new employees that belongs to the "HR Specialist" position.
- action: precondition property generally provides the conditions such as "In order to HR specialist can record a new employee to the HR application, Human Resources Specialist must have competency "Legal, Government and

jurisprudence”

HasUserRight {x, AccessInfo (insert, employee(x, (hasCompetency (y,CompetencyInformation(has, “Legal, Government and jurisprudence”))}

- action:effect

This property states the conditions that occur after the action is performed. For example, after printing 2 pages, there will be two less pages in the printer.

If properties like target, location, and actor of an action are instantiated to a specific instance then an action instance is described. Otherwise, if any one of the properties is an entity:Variable, it describes a set of actions. An example of a description for a set of actions is in Figure 29. Rei policy ontology graph is shown.

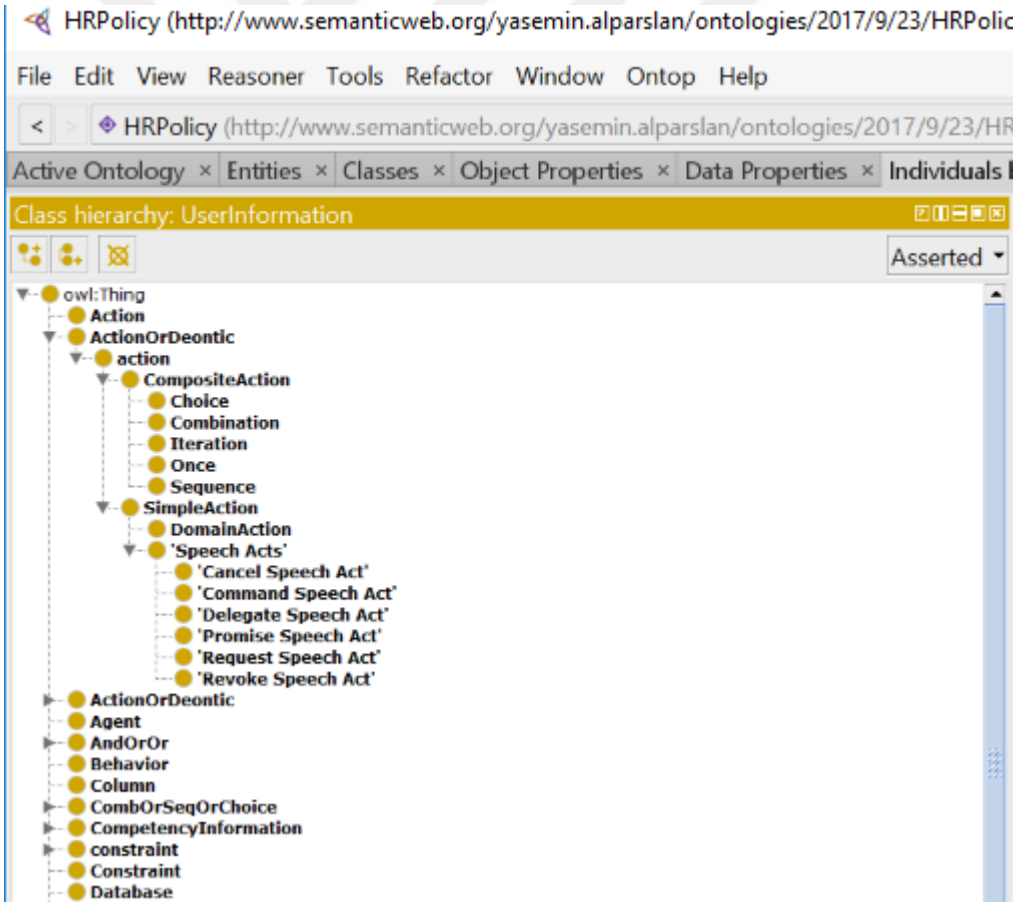


Figure 28. Speech act mapping between HR ontology and Rei Ontology

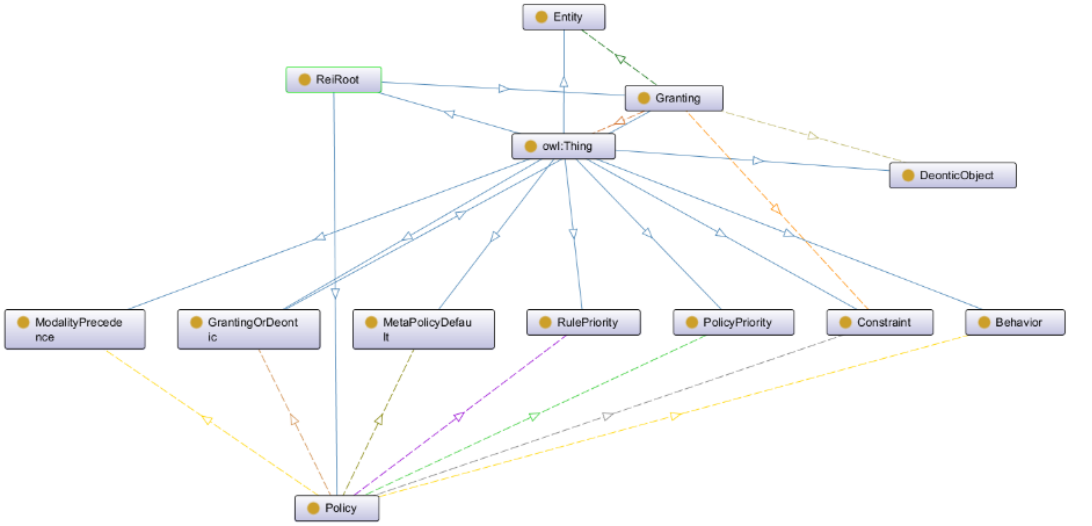


Figure 29. Rei Policy relations

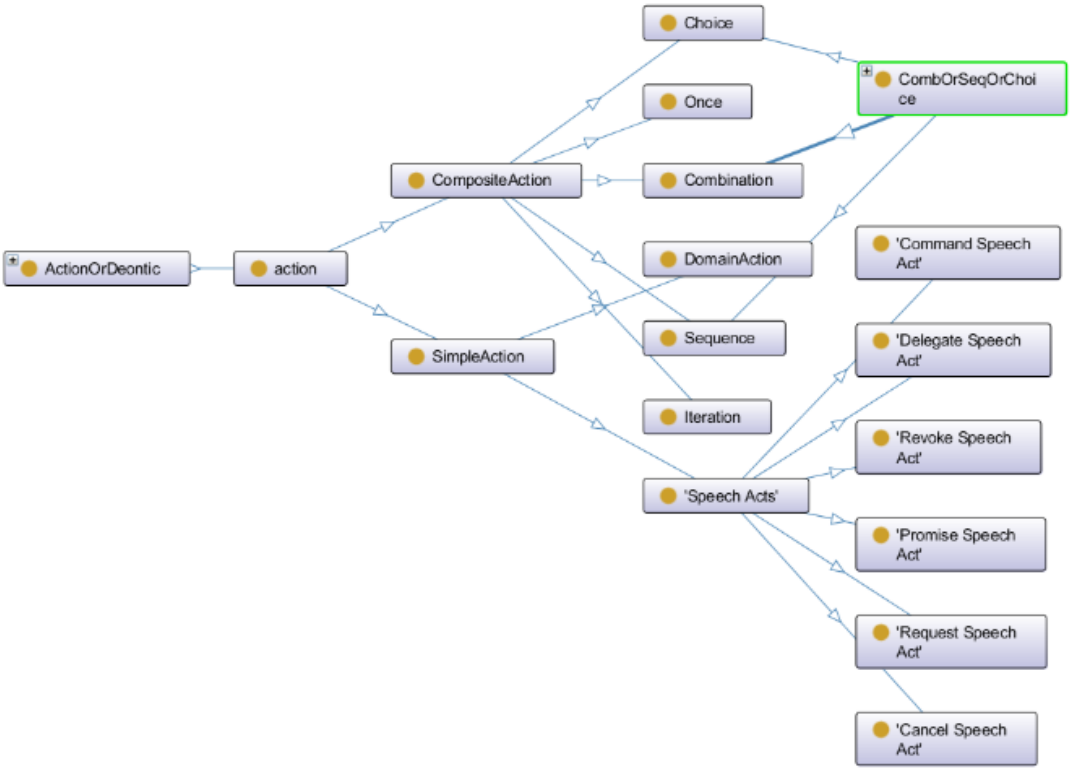


Figure 30. Rei Action relations

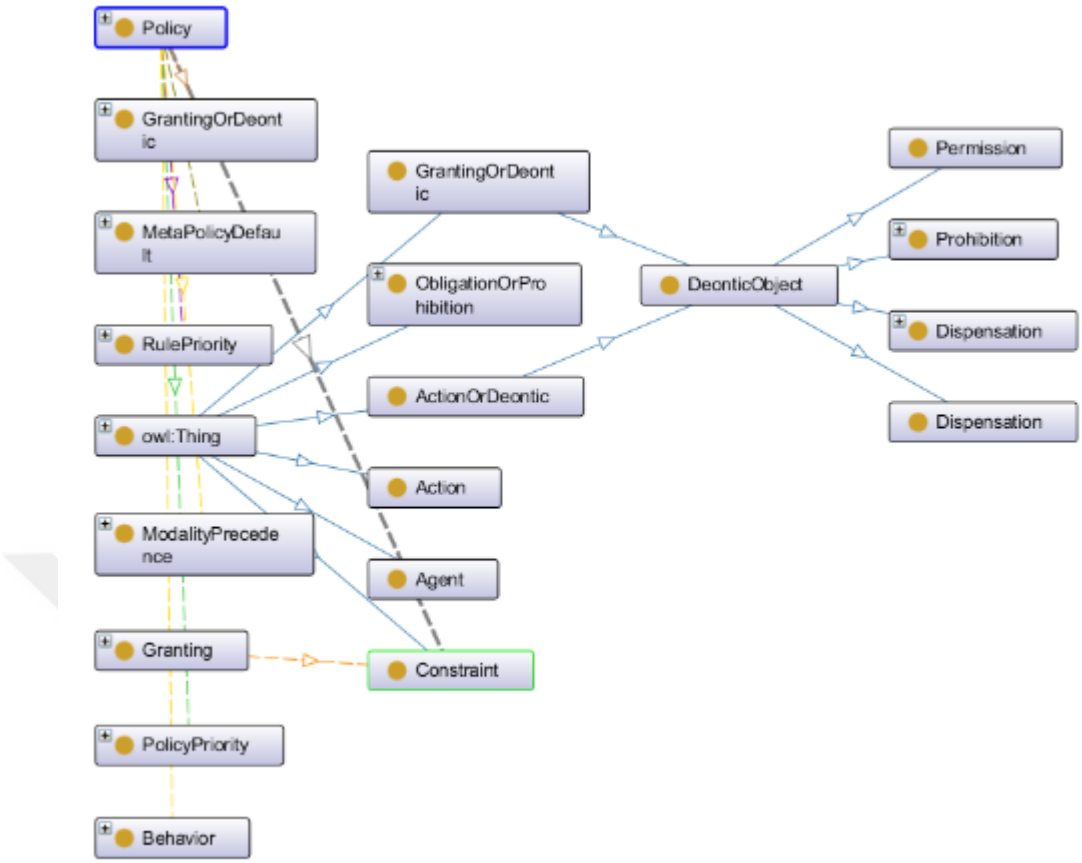


Figure 31. Rei Deontic relations

In future, there would be a profile definition. This will provide dynamic policy management. In this study, actors refer to positions. Which employee get that position, employee will gain the permission. But in the future, ontology can be designed as meta profile point of view.

3.5 Implementation: Policy Tool

With a user interface, policy creation, update, deletion, questioning on policy is aimed with a tool. The tool is developed in C# programming language. While coding the HR Policy Tool

3.5.1 C#

Program is developed in C# language using Visual Studio solution.

3.5.2 JenaDotNet

JenaDotNet can be downloaded by the link http://www.linkeddatatools.com/about-linked-data-tools. Jena .NET uses the IKVM virtual machine to bring the latest Jena package. Jena .NET is a fully-featured .NET port of the well-established open source Jena Semantic Web library, and features a robust RDF/OWL API, persistent storage support, RDF/XML/N3/N-triple support and a SPARQL engine for building your own semantic web applications, whilst taking advantage of the flexibility, power and convenience of the .NET framework. The Jena API Reference gives an overview of the Jena library's functionality with examples. LinkedDataTools.com will be also developing additional tutorials exploring the use of Jena .NET with the .NET framework. [45].

3.5.3 Semiodesk Trinity

Semiodesk Trinity is an application development platform for Microsoft .NET and Mono. It provides to build Linked Data and Semantic Web applications based on the RDF metadata easily. The Semiodesk Trinity platform allows to use ontology terms directly in the IDE, supporting code completion such as IntelliSense in Microsoft Visual Studio. Semiodesk Trinity provides a suitable way to implement value change notification through property attributes. This way, data objects can directly be manipulated from the UI. It provides a way to map RDFS/OWL classes to C# classes.

Core library can be added from Nuget Package. The Semiodesk.Trinity API has dependencies to dotNetRDF and OpenLink.Data.Virtuoso.

Semiodesk Trinity makes semantic object mapping by providing a way to map RDFS/OWL classes to C# classes,

3.5.4 Semiodesk Trinity Examples

In this section, basic C# code of semiodesk trinity processes for semantic web applications will be mentioned.

Creation of an ontology model

HR Policy Ontology, the FOAF. is used as data model.

First new HR Policy ontology should be known by Semiodesk Trinity. HR policy ontology should be extracted to the ontologies directory by using OntologySettings section in the App.config as in the Figure 32.

```
</Ontology>
        <Ontology Uri="http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/9/23/HRPolicy"
Prefix="owl">
          <FileSource Location="Ontologies\HRPolicy.owl" />

        </Ontology>
```

Figure 32. Ontology model creation in c#

In solution, there is Object Model folder which includes User, Actor, Action, Policy, Application model.

An example for creation of User object model

5 classes are created: User, Actor, Action, Policy, Application. In figure

```
namespace HRPolicy.User
{
   [RdfClass(HRPolicy.User)]
   class User : User
   {
      public void UserInformation(Uri uri) : base(uri) { }
      [RdfProperty(HRPolicy.UserInfo)]
      public string ApplicationInfo { get; set; }

      [RdfProperty(HRPolicy.ApplicationInfo)]
      public List<AccessInfo> Record { get; set; }

   }
```

Figure 33. User model creation from ontology in c#

Firstly it is needed to do some initialization. With the LoadOntologySettings() method we can import all ontologies from the current app.config file. In the case of the Virtuoso the ruleset is also created as in the Figure 34.

```
SemiodeskDiscovery.Discover();

IStore store = StoreFactory.CreateStoreFromConfiguration("virt0");

store.LoadOntologySettings();
```

Figure 34. Virtuoso rule set example

Then, a model can be created or opened. If the model exists, so I don't add the same information again. A model in RDF contains triples and is identified by a Uri. It can be used to group information of one domain together. A model creation is shown in Figure 35.

```
Uri modelUri = new
Uri("http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/9/23/HRPolicy ");

IModel model;

if (store.ContainsModel(modelUri))
{
  model = store.GetModel(modelUri);
  model.Clear();
}
else
{
  model = store.CreateModel(modelUri);

}
```

Figure 35. Model creation in c#

Reading RDF from URIs

By using dotnetRdf library from Semiodesk Trinity, RDF files can readed from a URI, with static UriLoader which provides a Load(IGraph g, String file) method as in Figure 36.

```
IGraph g = new Graph();

UriLoader.Load(g,                                                    new
Uri("http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/9/23/HRPolicy "));
```

Figure 36. Reading RDF from URI code part

3.5.5 Policy Tool Structure

Policy tool should be able to:

- Create a new action
- Create deontic policy definition: A new policy definition can be inserted. A policy definition includes HR domain, permission and prohibition rules, conflict solution.

- Update existing deontic policy: All part of the policy can be updated.
- Create new access info to actor

User interface of the HR Policy Manager shown in Figure 37.

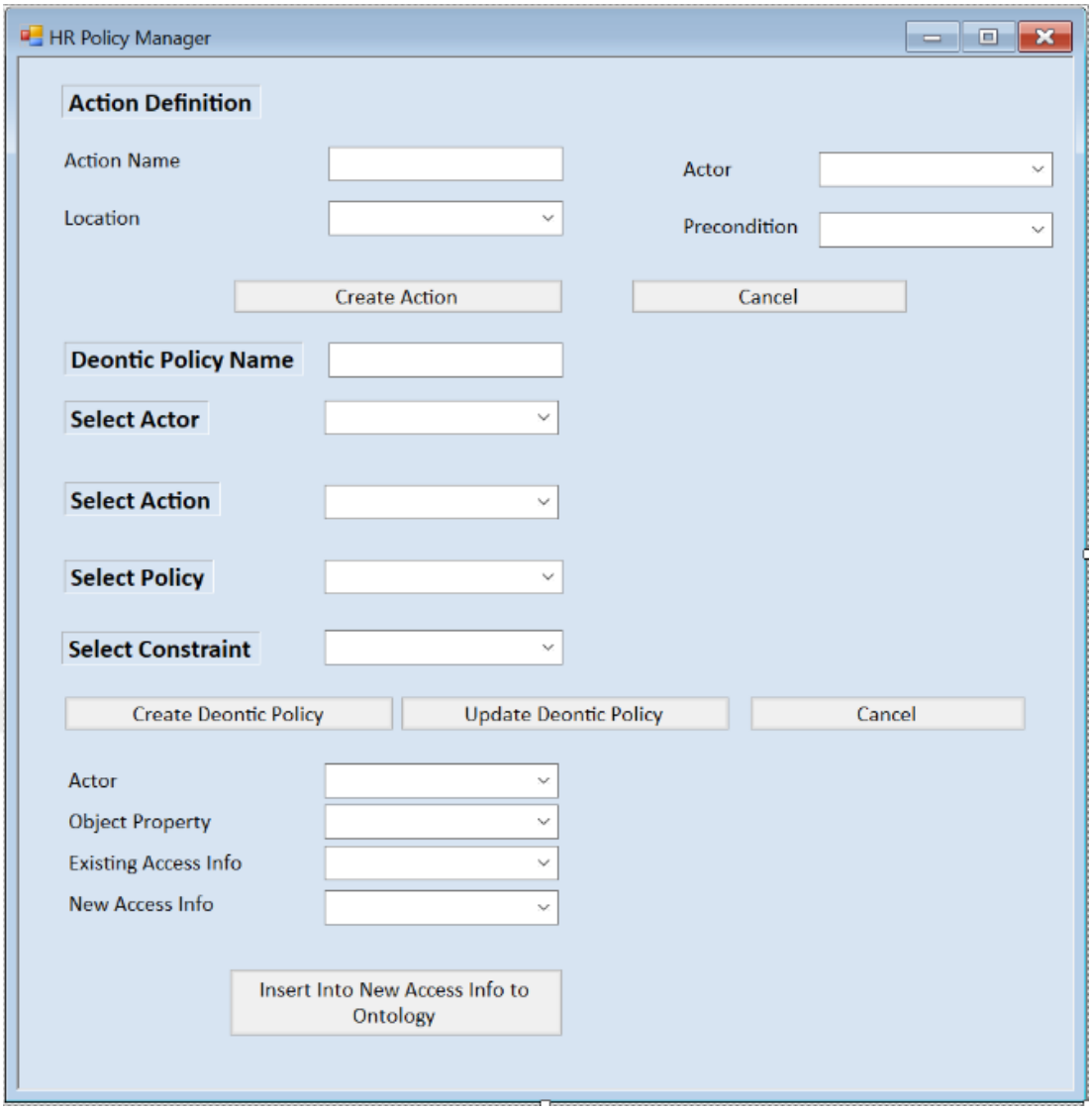


Figure 37. HR Policy Manager User Interface

Hybrid Base access control is created on the base of Figure 38.

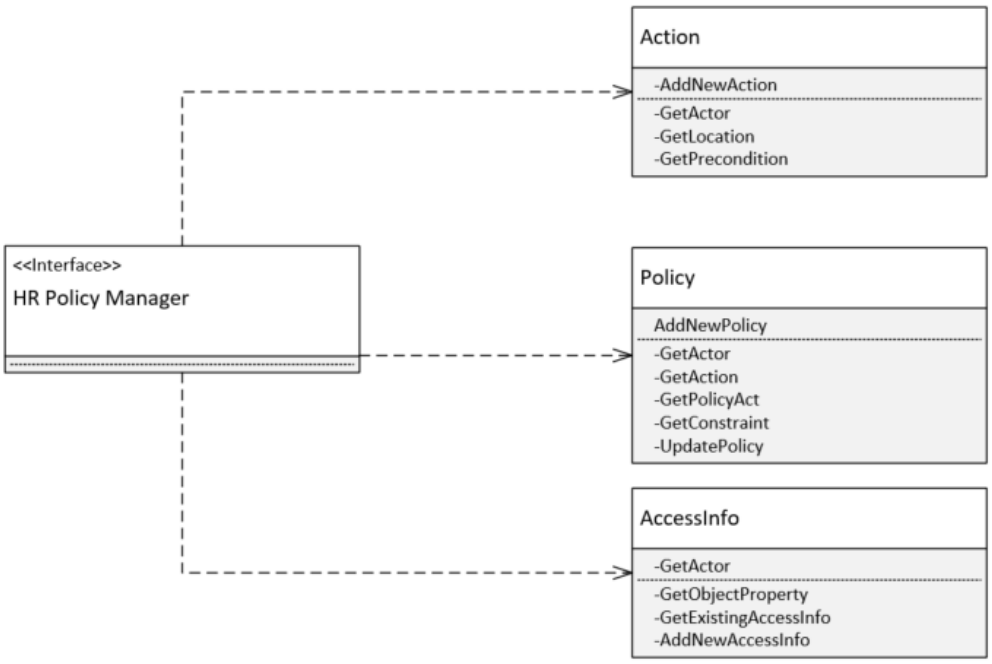


Figure 38. HR Policy Manager User interface parts

Creation of the access control mechanism can have divided in three parts: Action Definition, Policy Definition, AccessInfo Modification

3.5.5.1 Action Definition

Action definition is required. Create Action includes action name, actor, location, precondition as in the Figure 39.

In HRPolicy.owl, all definitions are done, so Actor selection is readed from rdf file and this will fill the actor combobox.

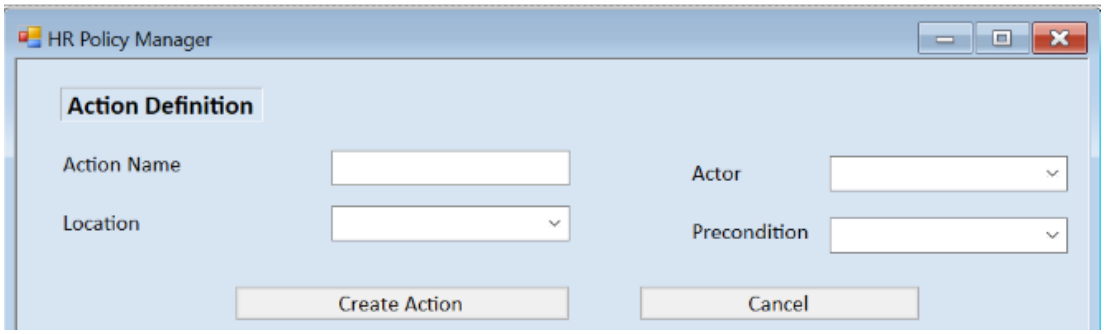


Figure 39. Action Definition User Interface

Actor: list includes, person class of subclasses: Employee and Candidate.

Precondition: definitions are listed from rdf. Precondition examples can be consisting of Simple Constraint or Complex Constraint.

- Simple Constraint of precondition examples are listed below:

  ✓ “Employee” hasPosition “Trainee”
  ✓ “Employee” hasCompetency “Legal, Government and jurisprudence”
  ✓ “Employee” hasPosition “HRSpecialist”
  ✓ “Employee” hasPosition “ITSpecialist”

- Complex Constraints of precondition examples are listed below:

  ✓ “Employee” hasEducationalnfo “EducationalInformation” AND “EducationalInformation” hasColumn “LISANS1BOLUM”

As in the example Hybrid data model is used because educational information reduced from SQL relational database.

  ✓ “Employee” hasApplicationAccess “AccessInfo” AND “AccesInfo” hasColumn “KAYIT”.
    - This precondition can be used for access right with several types as in Figure 40. In complex constraint, access rights can be used as in Figure 40.

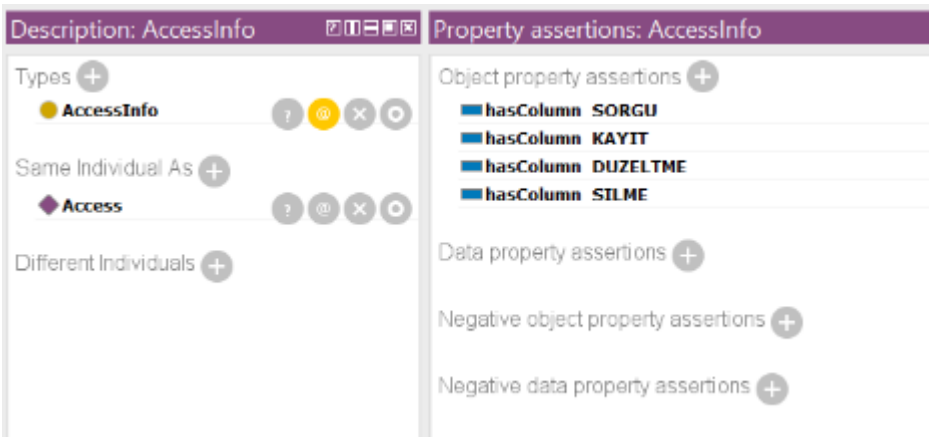


Figure 40. AccessInfo HasColumn objects maps the deontic objects in Rei Policy Language

Location: In this study, location means HR application'modules and programs. As it is told previous parts, HR application has modules and programs and they can be considered to map location of action definition.

"Employee" hasApplicationAccess "ApplicationInfo". Application Info instance is HRWEB.

"ApplicationInfo" has instance as "ModulInfo"

"ModulInfo" hasColumn "MODUL_NO"

"ApplicationInfo" has instance as "ProgramInfo"

"ProgramInfo" hasColumn "PROGRAM_NO"

HR application has modules such as Employee, Recruitment, Organizational Management and modules are mapping to module no as it shown in Table 7.

Table 7. Module names of HR applications

MODUL_NO	Modul Name
400	Definitions (Web)
401	Employee (Web)
404	User (Web)
407	Recruitment (Web)
411	OrganizationalManagement (Web)

3.5.5.2 Deontic Policy Definition

New deontic policy definition requires policy name, actor, action, policy, constraint definition.

Actor, Action, Policy and Constraint definition are read from HRPolicy ontology. And it maps to rei: policy. It writes to rei: policy part of HRPolicy ontology.

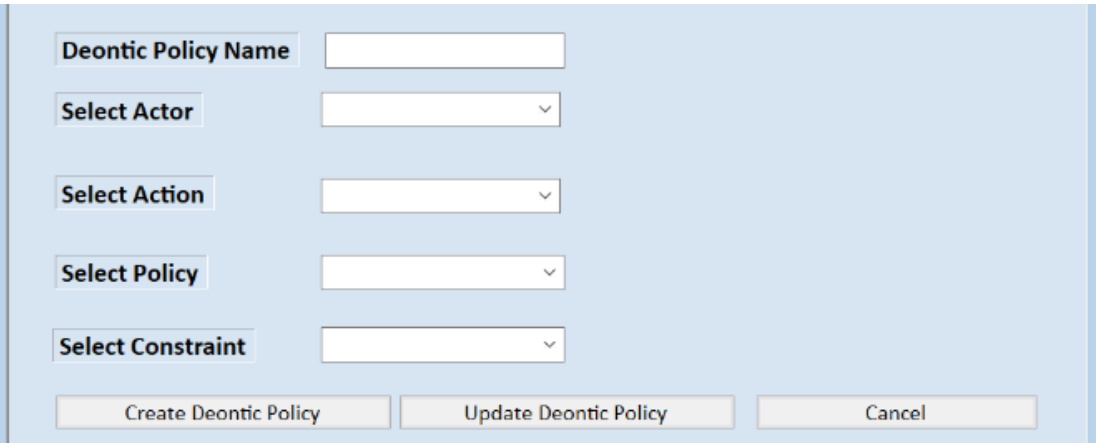


Figure 41. Deontic Policy User Interface

3.5.5.3 New Access Info to Ontology

Actor, Object Property, Existing Access Info are listed from the HRPolicy.Ontology.

Access Info has column KAYIT, SORGU, SILME and DUZELTME as in Figure 42.

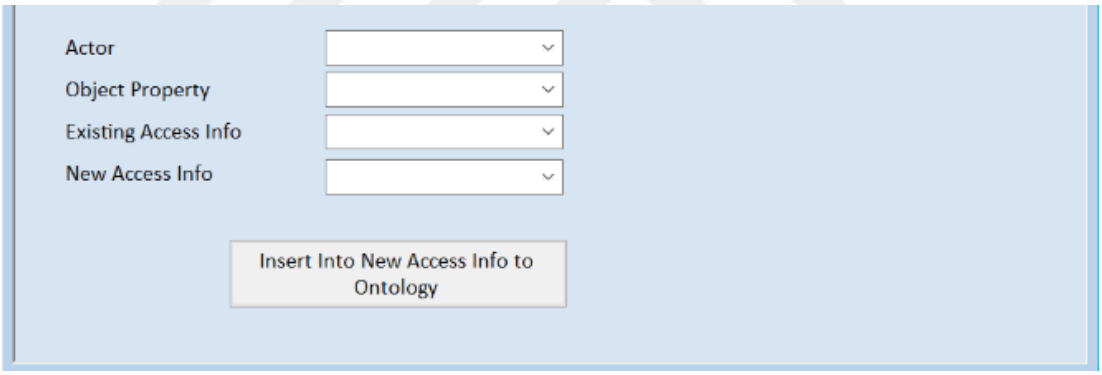


Figure 42. Access Control Process

3.5.5.4 Policy Granting

Policy approval which also means “Policy Granting” is done automatically when “Insert into New Access Info to Ontology” button.

This process is the final process and is done automatically done when the all information is saved with the “Insert into New Access Info to Ontology” button.

3.5.5.5 Comparison of HR RBAC and HR HAC

In HR RDBMS user rights on the module of applications is recorded on to the database as Figure 43.

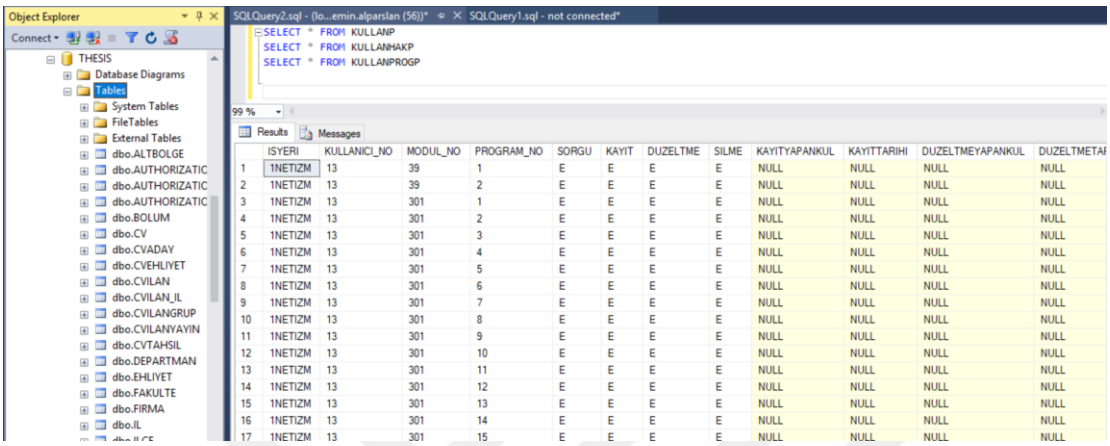


Figure 43. HR Application user rights on database

Access Control Example: In order to HR specialist can record a new employee to the HR application, Human Resources Specialist must have competency “Legal, Government and jurisprudence”

The comparison of the access control mechanisms between RBAC and HAC is expressed below.

RBAC in HR

In RBAC, sql query should be like in Figure 44.

```
UPDATE KULLANHAKP SET KAYIT='E' WHERE ISYERI+CAST(KULLANICI_NO AS
VARCHAR) IN(SELECT KP.ISYERI+CAST(KP.KULLANICI_NO AS VARCHAR) FROM
KULLANPROGP KP
LEFT OUTER JOIN KULLANP K ON K.ISYERI=KP.ISYERI AND
K.KULLANICI_NO=KP.KULLANICI_NO
LEFT OUTER JOIN OZLUK O ON O.OZLUKID=K.OZLUKID
LEFT OUTER JOIN ORGGECMIS OG ON OG.OZLUKID=O.OZLUKID
```

```
LEFT OUTER JOIN POZISYON POZ ON POZ.POZISYON_KOD=OG.POZISYON_KOD
LEFT OUTER JOIN PROFIL P ON P.OZLUKID=O.OZLUKID
LEFT OUTER JOIN YETKINLIK Y ON Y.YETKINLIKID=P.YETKINLIKID
LEFT OUTER JOIN NITELIK N ON N.NITELIKID=Y.YETKINLIKID
WHERE N.KISAACIKLAMA='Legal, Government and jurisprudence'
AND POZ.POZISYON_ISIM='HR Specialist'
AND KP.MODUL_NO=400
AND KP.PROGRAM_NO=11)
```

Figure 44. Access Control asssignment in RBAC

In Figure 43, there is an update query of a user insert right to Employee Module for an employee who has position “HR Specialist” and who has competency like ‘Legal, Government and jurisprudence’. Query Statistics are shpwn in Figure 45.

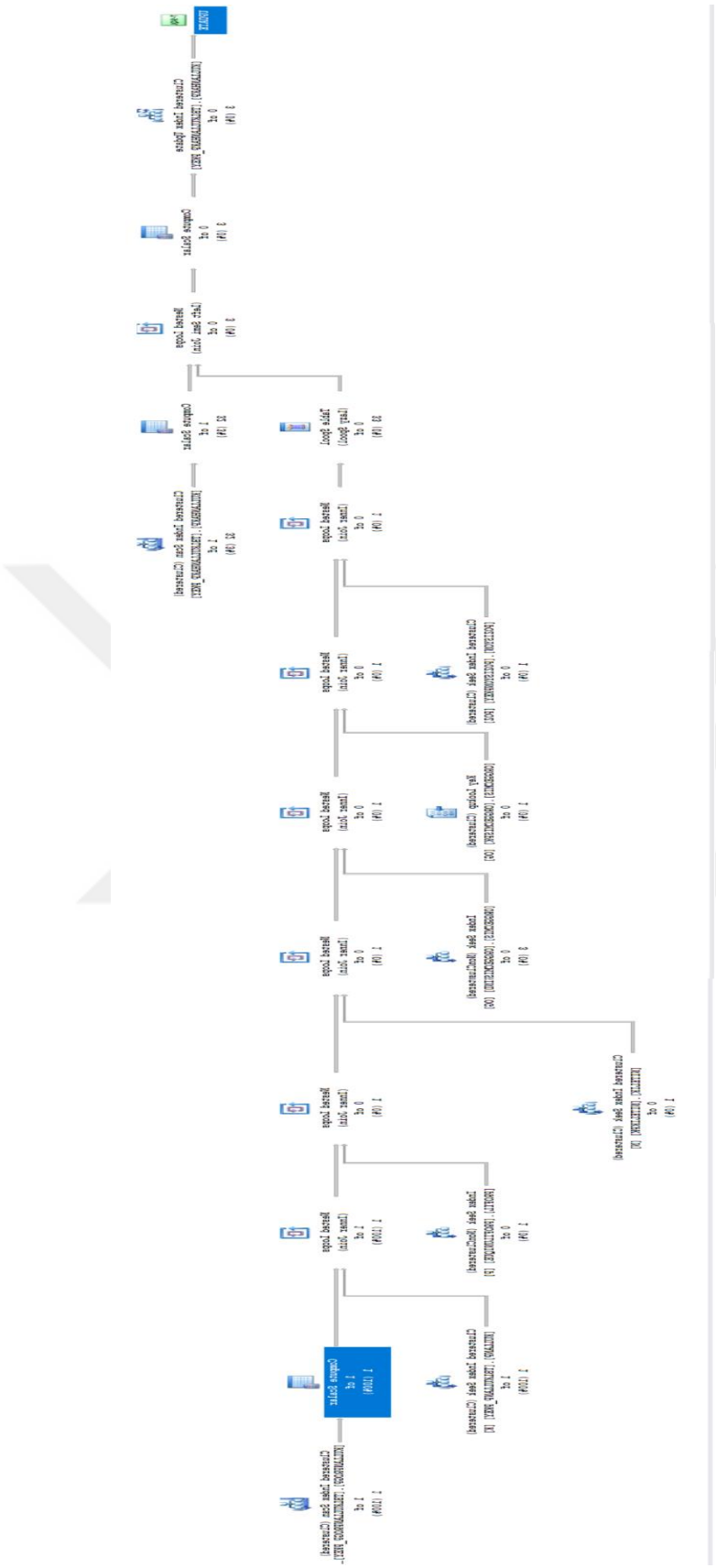


Figure 45. SQL Query Statistics for giving an employee to insert right to Employee

Studied database is only a sample for HR management. For example, there is only 50 employees in tables or 30 position definition in tables. But in real life, companies may have thousands of employees. Number of join between tables has an extremely cost for big data. Transactions, data transfer, insert, update, delete operations on database become a critical issue. With query optimization techniques, the query in Figure 43 can be improved but it wouldn't be enough for the best.

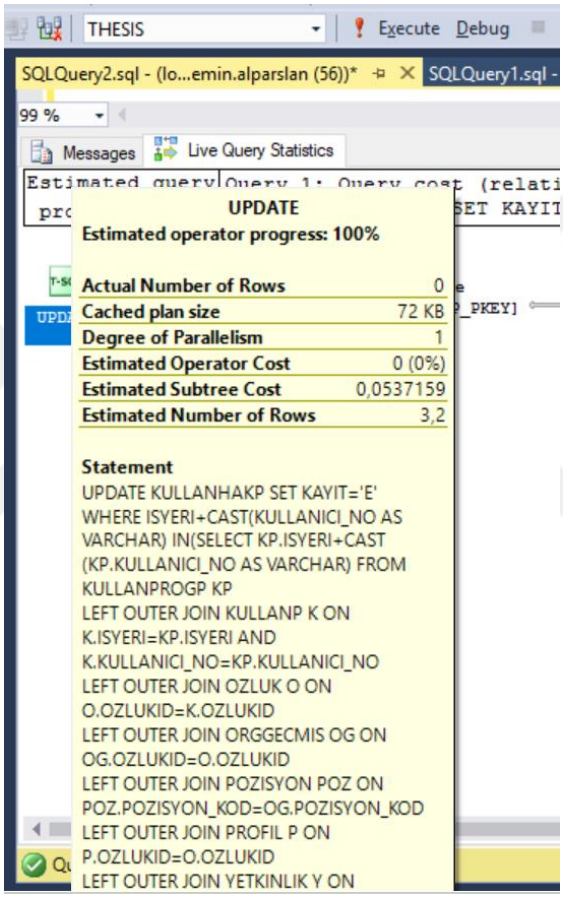


Figure 46. Update Query Statistics

The SQL Server query optimizer generates a number of physical plan alternatives from a logical requirement expressed in T-SQL. When full cost-based optimization is required, a cost is assigned to each iterator in each alternative plan, and the plan with the lowest overall cost is ultimately selected for execution. The estimated cost values mean whether the query is I/O intensive or CPU intensive. [45]

In RBAC, sql queries are used for the assignment of the access rights between employees and modules. Its performance cost may take too much.

HAC in HR

Hybrid Access Control provides ontological access control using SPARQL.

HasUserRight {x, AccessInfo (insert, employee(x, (hasCompetency (y,CompetencyInformation(has, "Legal, Government and jurisprudence"))}

```
SELECT ?UserInfo

WHERE

?Who <Person#Employee> ?hasUserInfo;

?Who <Position#column> "HRSpecialist"

?Who <hasCompetency#column> "Legal, Government and jurisprudence"

 ?Who <hasApplicationInfo#column ?MODUL_NO "400"

?Who <hasApplicationInfo#column ?PROGRAM_NO "11"

}
```

Figure 47. Access Control Assignmnet in HAC

At this point, the answer of the question which is "what can be done with SPARQL that can not be done with SQL" should be understood.

Both of SQL and SPARQL languages provides the user access to create, combine, and consume structured data. SQL does this by accessing and joining tables in relational databases, and SPARQL does this by accessing a web of Linked Data.

SQL Structure	SPARQL Structure
SELECT <attribute list> FROM <table list> WHERE <test expression>	SELECT <variable list> WHERE {<graph pattern> }

Figure 48. SPARQL and SQL Structure

Every RDF statement has subject, predicate and object. There is no concept in RDF corresponding to SQL’ s NULL as there is no RDF requirement corresponding to SQL’ s structural constraint that every row in a relational database must conform to the same schema. RDF is a post-Web language so this is the basic difference between them. RDF allows one to use web identifiers for the entities we want to describe. RDF data is mergable in a way which for relational data would require an intermediate process of mapping the databases and assuring that no tables used the same column name to mean different things [46].

A SPARQL query selecting each user information, in the SQL; the result of the query is a list of rows.SQL databases are respositories of data, with a set of tables populated by rows of data. SQL queries operate over a given database. SPARQL services vary as in Figure 48 in whether or not they have a pre-determined RDF database [46]. According to the comparison cost of HAC is less than RBAC.

```
SELECT ?UserInfo

FROM<
http://www.semanticweb.org/yasemin.alparslan/ontologies/2017/9/23/HRPolicy >

WHERE { … }
```

Figure 49. Sparql example with web source

CHAPTER 4

CONCLUSIONS AND FUTURE RESEARCH

As a conclusion, hybrid data which is occurred by ontology and database together needs to access control, and hybrid based access control brings a solution for access control in this condition. Ontology that is created in this study is not only relational owl. Also, HAC has quite different from ROWLBAC. Because, RBAC is not converted in to OWL. Within the purview of hybrid data ontology, every concept that has a meaning for internal operations can be introduced. A tool software is developed in accordance to proposed software infrastructure.

Completed studies can be grouped by under titles below:

 Developing an HR Ontology: This part is also can be considered as contribution, because used HR ontology is not taken from any where, it is generated from relational database but differently having additions according to different cases. HR Relational database is also optimized by new HR ontology using relational owl. But new HR ontology is further than relational owl.

 Developing Policy Ontology: Using Rei policy language, hybrid access control is provided.

 Application Developing: Using Semiodesk Trinity libraries, HR Policy Manager tool is developed in C# language. C# application and intregration with the ontology that is created from Protégé is provided. Application can change access control from relational database to new access control in ontology.

 Query Developing: Using SPARQL query language, HR Policy Manager tool get results from the code parts.

 Comparison of Access Control Mechanisms: RDF query cost is smaller than RDBMS query cost.

A hybrid access control is provided by using an HR ontology within the OWL language and HR relational database in SQL. Queries that is used for access control in HR ontology

is taken by SPARQL language. A tool is developed in C# language by using OWL which is created as ontological data by using dotnetRDF libraries in Semiodesk Trinity. Tool is developed in Microsoft Visual Studio platform. Hybrid Access Control is provided in C# programming language. For future work this study can be improved with NoSQL, MongoDB.

REFERENCES

1. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.
2. Gómez-Pérez, A., Ramírez, J., & Villazón-Terrazas, B. (2007). An ontology for modelling human resources management based on standards. In *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 534-541). Springer Berlin/Heidelberg.
3. Dorn, Jürgen, Tabbasum Naz, and Markus Pichlmair. "Ontology Development for Human Resource Management." Proceedings of the 4th International Conference on Knowledge Managements. 2007.
4. http://www.hropenstandards.org [Erişim Tarihi:20 Mayıs 2015]
5. Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274.
6. Can, Ö., Ünalır M.O., "Ontoloji Tabanlı Erişim Denetimi."Pamukkale University Journal of Engineering Sciences 16.2 (2010).
7. Maniu, George, and Ionela Maniu. "A human resource ontology for recruitment process." Review of General Management 2 (2009): 12-18
8. Pernici, Barbara, Paolo Locatelli, and Clementina Marinoni. "The eCCO system: an eCompetence management tool based on semantic networks." On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. Springer Berlin Heidelberg, 2006.
9. IEEE 1484.20.1/Draft 3, Draft Standard for Learning Technology – Standard for Reusable Competency Definitions, Learning Technology Standards Committee of the IEEE Computer Society (08 March 2006)
10. Dittmann, L., and Stephan Zelewski. "Ontology-based skills management."Proc. of the 8th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2004). Vol. 4. 2004.
11. Mochol, Malgorzata, Elena Paslaru, and Bontas Simperl. "Practical guidelines for building semantic erecruitment applications." International Conference on Knowledge Management, Special Track: Advanced Semantic Technologies (AST'06). 2006.

12. Kirrane, Sabrina. Linked Data with Access Control. Diss. National University of Ireland, Galway, 2015.
13. Campbell, R. H.; Liu, Z.; Mickunas, M. D.; Naldurg, P.; Yi, S., Seraphim: Building Dynamic Interoperable Security Architecture For Active Networks, Open Architectures and Network Architectures and Network Programming, 2000 (OPENARCH 2000), 2000, pp55–64
14. Samarati, P.; Jajodia, S., Data Security, from Webster, J.G., Wiley Encyclopedia of Electrical and Electronics Engineering, John Wiley & Sons, 1999
15. Yuan, Eric, and Jin Tong. "Attributed based access control (ABAC) for web services." Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE, 2005.
16. Sandhu, R. S., 1998. Role-based access control. Advances in Computers, Elsevier 46, 237 – 286. URL http://www.sciencedirect.com/science/article/pii/S0065245808602065
17. Anderson, A. (2005). Core and hierarchical role based access control (RBAC) profile of XACML v2. 0. OASIS Standard, 2005.
18. Sandhu, Ravi, David Ferraiolo, and Richard Kuhn. "The NIST model for role-based access control: towards a unified standard." ACM workshop on Role-based access control. Vol. 2000. 2000.
19. Can, Ö., Bursa, O., Ünalır M.O., "Personalizable Ontology Based Access Control." Gazi University Journal of Science 23.4 (2010): 465-474.
20. Mohammad, A., et al. "Ontology–Based Access Control Model for Semantic Web Services." Journal of Information and Computing Science 6.3 (2011): 177-194
21. Finin, Tim, et al. "Using owl to model role based access control." Organization (2008): 125.
22. Lorch, Markus, Dennis Kafura, and Sumit Shah. "An XACML-based policy management and authorization service for globus resources." Proceedings of the 4th International Workshop on Grid Computing. IEEE Computer Society, 2003.
23. Kagal, Lalana. "Rei." (2002)
24. Dean, Mike, et al. "OWL web ontology language reference." W3C Recommendation February 10 (2004).

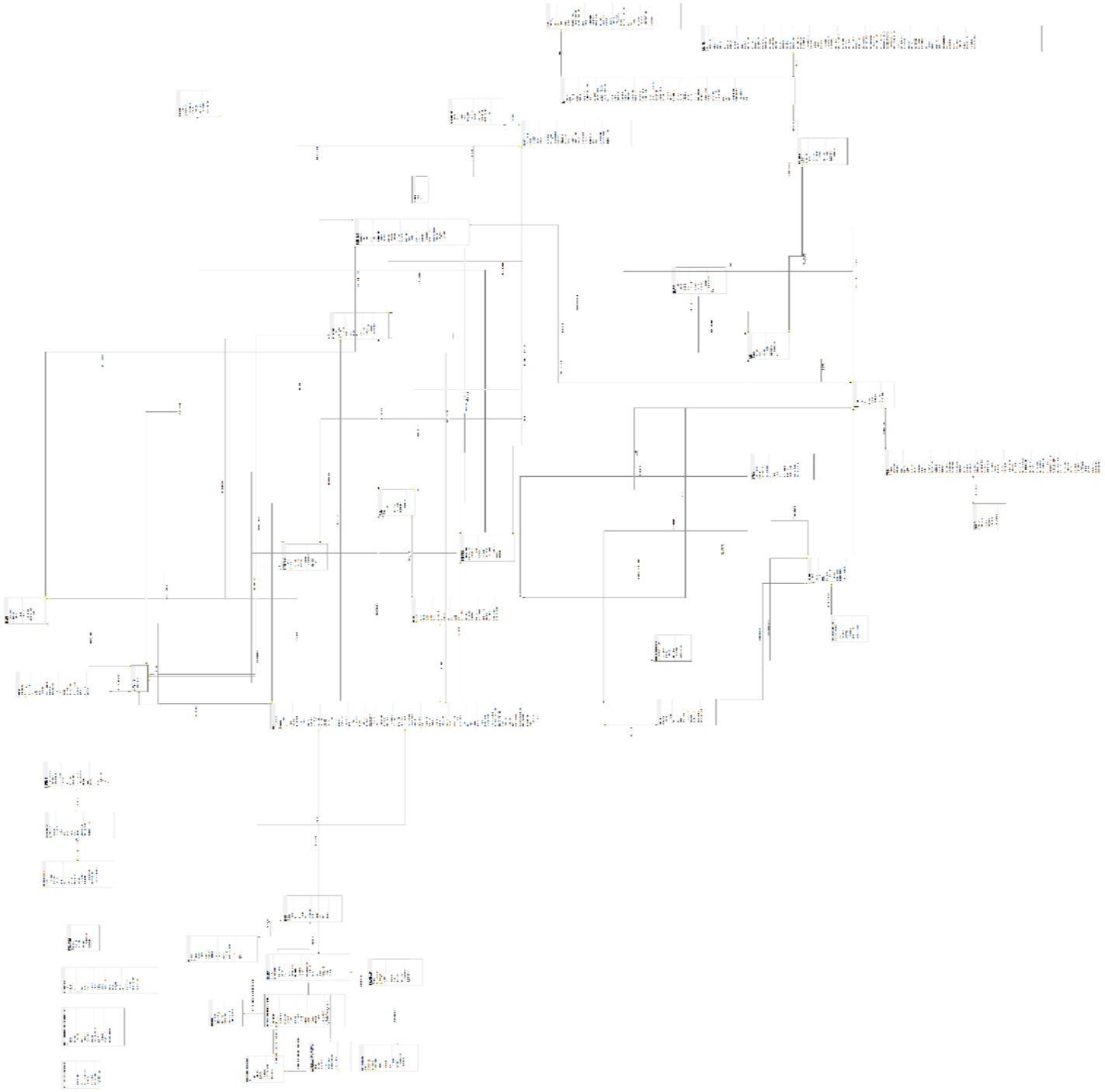
25. He, Lijuan, et al. "Design of policy language expression in SIoT." Wireless and Optical Communication Conference (WOCC), 2013 22nd. IEEE, 2013.
26. Laclavık, M. (2007). RDB2Onto: Relational database data to ontology individuals mapping. In Proceeding of ninth international conference of informatics.
27. Astrova, I. (2009). Rules for mapping SQL relational databases to OWL ontologies. Metadata and Semantics, 415-424.
28. Alparslan, N. Y., Komesli, M., Ünalir, M. O., & Can, Ö. (2015). Melez Erişim Denetimi için bir Mimari Önerisi: İK Uygulaması Örneği. In UYMS,2015.
29. Cullot, N., Ghawi, R., & Yétongnon, K. (2007, June). DB2OWL: A Tool for Automatic Database-to-Ontology Mapping. In SEBD (pp. 491-494).
30. Santoso, H. A., Haw, S. C., & Abdul-Mehdi, Z. T. (2011). Ontology extraction from relational database: Concept hierarchy as background knowledge. Knowledge-Based Systems, 24(3), 457-464.
31. Čerāns, K., & Būmans, G. (2011). RDB2OWL: a RDB-to-RDF/OWL mapping specification language. Information Systems, 139-152.
32. Sahoo, S. S., Halb, W., Hellmann, S., Idehen, K., Thibodeau Jr, T., Auer, S., & Ezzat, A. (2009). A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Report, 1, 113-130.
33. Sloman, M. (1994). Policy driven management for distributed systems. Journal of network and Systems Management, 2(4), 333-360.
34. Kagal, L. (2002). Rei.
35. Can, Ö. (2009). Anlamsal web için kişiselleştirilebilir ontoloji tabanlı erişim denetimi ve politika yönetimi (Doctoral dissertation, Ege Üniversitesi).
36. Macfie, A. (2014). Semantic role-based access control (Doctoral dissertation, University of Westminster).
37. http://rei.umbc.edu/ (Access Date: 02.07.2017)
38. Toninelli, A., Bradshaw, J. M., Kagal, L., & Montanari, R. (2005, September). Rule-based and ontology-based policies: Toward a hybrid approach to control agents in pervasive environments. In Semantic Web and Policy Workshop (pp. 42-54).
39. Tonti, G., Bradshaw, J., Jeffers, R., Montanari, R., Suri, N., & Uszok, A. (2003). Semantic Web languages for policy representation and reasoning: A

comparison of KAoS, Rei, and Ponder. The semantic web-ISWC 2003, 419-437.

40. Kagal, L., Finin, T., & Joshi, A. (2003, June). A policy language for a pervasive computing environment. In Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on (pp. 63-74). IEEE.
41. R Yavatkar, D Pendarakis, and R Guerin, "A Framework for Policy-based Admission Control", IETF Informational Standard, RFC 2753, January 2000
42. Anderson, A. H. (2004, June). An introduction to the web services policy language (wspl). In Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on (pp. 189-192). IEEE.
43. OASIS Committee Specification, http://www.oasis-open.org/committees/download.php/4103/cs-xacml-specification- 1.1.doc, 24 July 2003.
44. http://www.linkeddatatools.com/downloads/jena-net(Access Date: 01.09.2017)
45. White, Paul. "Inside the Optimizer: Plan Costing" http://sqlblog.com/blogs/paul_white/archive/2010/09/01/inside-the-optimizer-plan-costing.aspx (Access Date: 01.12.2017)
46. "SPARQL vs SQL" https://www.cambridgesemantics.com/blog/semantic-university/learn-sparql/sparql-vs-sql/ (Access Date: 01.12.2017)

APPENDIX 1 – Database Diagram

Database diagram and relations between tables are shown in database diagram.



APPENDIX 2 – Column names of each table

Table Name	Column Name
PROFIL	PROFILID
	OZLUKID
	YETKINLIKID
	BASTAR
	BITTAR
	AKTIF
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
CV	KULLANICI_NO
	CVID
	CVKAYNAK
	KARIYERHEDEFI
	ISVERENARAMAKELIMESI
	CINSIYET
	MEDENIHAL

	ASKERLIKDURUMU
	ASKERLIKDURUMTARIHI
	ASKERLIKDURUMNEDENI
	SIGARA
	ENGELLI
	ENGELLIKATEGORI
	ENGELLIDERECE
	ENGELLIACIKLAMA
	SURELIENGELLI
	ENGELLIBASTAR
	ENGELLIBITTAR
	ESKIHUKUMLU
	HUKUMKATEGORI
	HUKUMACIKLAMA
	TERORMAGDURU
	TERORMAGDURUACIKLAMA
	BITIRDIGITAHSILI
	DEVAMETTIGITAHSILI
	BILGISAYARBILGISI
	BEKLENENHAKLAR

	SEYAHAT_EDEBILIR
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	AKTIF
	HOBI
	RESIM
	TOPLAMCALISMAYILI
	CALISMADURUMU
	UCRETBEKLENTISI
	SILINENCV
	ADAYID
KULLANPROGP	ISYERI
	KULLANICI_NO
	MODUL_NO
	PROGRAM_NO
	SORGU
	KAYIT
	DUZELTME
	SILME

	KAYITYAPANKUL
	KAYITTARIHI
	DUZELTMEYAPANKUL
	DUZELTMETARIHI
TAHSIL	OZLUKID
	ILKOKULISMI
	ILKMEZUN
	LISETIPI
	LISEBOLUMU
	LISEOKULISMI
	LISEBITISYILI
	LISEDILI
	LISEILI
	LISEMEZUN
	LISENOTSISTEMI
	LISEDERECE
	LISANS1TIPI
	LISANS1UNVTIPI
	LISANS1OGRTIPI
	LISANS1UNIVERSITE
	LISANS1FAKULTE

	LISANS1BOLUM
	LISANS1DILI
	LISANS1ILI
	LISANS1BURS
	LISANS1BASYIL
	LISANS1BITYIL
	LISANS1MEZUN
	LISANS1NOTSISTEMI
	LISANS1DERECE
	LISANS1ACIKLAMA
	YUKSEKBOLUM
	YUKSEKILI
	YUKSEKBASYIL
	YUKSEKBITYIL
	YUKSEKMEZUN
	YUKSEKNOTSISTEMI
	YUKSEKDERECE
	YUKSEKACIKLAMA
	DOKTORAUNVTIPI
	DOKTORAUNIVERSITE
	DOKTORABOLUM

	DOKTORAILI
	DOKTORABITYIL
	DOKTORAMEZUN
	DOKTORAACIKLAMA
	LISEHAZIRLIKEGITIMI
	LISANS1HAZIRLIKEGITIMI
	YUKSEKHAZIRLIKEGITIMI
	DOKTORAHAZIRLIKEGITIMI
	YUKSEKTEZ
	DOKTORAENSTITU
	YUKSEKENSTITU
	LISEULKESI
	LISANS1ULKESI
	YUKSEKULKESI
	DOKTORAULKESI
	YAYINLAR
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
KULLANP	ISYERI

	KULLANICI_NO
	KULLANICI_ISIM
	ISIM_SOYAD
	SIFRE
	ENSON_SIFRETAR
	ENSON_LOGINTAR
	SIFRE_DEGIS
	KAYITYAPANKUL
	KAYITTARIHI
	DUZELTMEYAPANKUL
	DUZELTMETARIHI
	ADMINMI
	CARDID
	HRADMIN
	PAYROLLADMIN
	OZLUKID
	SSOUSERID
POZISYON	ISLETME_KODU
	POZISYON_KOD
	POZISYON_ISIM
	GOREV_TANIMI

	KADEMESI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	INCKEYNO
ILCE	ILCEID
	ILCE_ISIM
	IL_KOD
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	ILCE_KOD
SICIL	ISYERI
	SICILNO
	GIRISTAR
	CIKISTAR
	AYRILIS_KODU
	AYRILIS_NEDENI
	GOREV_KODU

	KAYITTAR
	KAYITYAPAN
	DUZELTMETAR
	DUZELTMEYAPANKUL
	ILKISYERI
	ILKSICILNO
	OZLUKID
IL	IL_KOD
	IL_ISIM
	ULKEID
	TELKODU
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
AUTHORIZATIONGROUPUSER	SIRKETKODU
	KULLANICINO
	AUTHGROUPID
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMEYAPANKUL

	DUZELTMETARIHI
	Id
SICILGECMIS	ISYERI
	SICILNO
	YIL
	AY
	AYICI
	UCRETI
	NETI
	MEDENIHAL
	GIRISTAR
	CIKISTAR
	AYRILIS_KODU
	AYRILIS_NEDENI
	TAHSILI
	GOREVI
	GOREV_KODU
	UNVAN
	EMAIL
	KAYITYAPANKUL
	KAYITTARIHI

	DUZELTMEYAPANKUL
	DUZELTMETARIHI
AUTHORIZATIONGROUPOBJECT	ID
	AUTHGROUPID
	OBJECTTYPEID
	ALTBOLGE_KOD
	BOLUM_KOD
	DEPT_KOD
	POZISYON_KOD
	ORGULKEID
	ORGILCEID
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMEYAPANKUL
	DUZELTMETARIHI
FIRMA	ISYERI
	UNVAN
	ILCE
	IL
	TELEFON
	GIZLI

	KURULUSTAR
	ISLETME_KODU
	MERKEZ
	EPOSTA
	WEB
	MERSISNO
AUTHORIZATIONGROUP	ID
	CODE
	DESCRIPTION
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMEYAPANKUL
	DUZELTMETARIHI
CVADAY	KULLANICI_NO
	ADAYID
	ADI
	SOYADI
	CINSIYET
	MEDENIHAL
	ASKERLIKDURUMU
	ASKERLIKDURUMTARIHI

	ASKERLIKDURUMNEDENI
	SIGARA
	ENGELLI
	ENGELLIKATEGORI
	ENGELLIDERECE
	ENGELLIACIKLAMA
	SURELIENGELLI
	ENGELLIBASTAR
	ENGELLIBITTAR
	ESKIHUKUMLU
	HUKUMKATEGORI
	HUKUMACIKLAMA
	TERORMAGDURU
	TERORMAGDURUACIKLAMA
	RESIM
	AKTIF
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL

	EMAIL
	ADAY_DURUM
	SEYAHAT_EDEBILIR
KULLANHAKP	ISYERI
	KULLANICI_NO
	MODUL_NO
	SORGU
	KAYIT
	DUZELTME
	SILME
	KAYITYAPANKUL
	KAYITTARIHI
	DUZELTMEYAPANKUL
	DUZELTMETARIHI
OZLUK	ISLETME_KODU
	OZLUKID
	OZLUKKODU
	ADI
	SOYADI
	ILKSOYADI
	BUNYE_GIRTAR

	SSK_NO
	SSK_DURUM
	ALTBOLGE_KOD
	ORGULKEID
	ORGIL_KOD
	ORGILCEID
	BOLUM_KOD
	DEPT_KOD
	POZISYON_KOD
	GOREV_TANIMI
	AMIRID
	AMIRLINE
	MAVIBEYAZYAKA
	CINSIYET
	MEDENIHAL
	ASKERLIKTIP
	TECIL_TERHIS_TAR
	MUAFNEDEN
	ENGELLIDERECE
	ENGELLIKATEGORI
	ENGELLIORAN

	ENGELLIACIKLAMA
	SURELIENGELLI
	ENGELLIBASTAR
	ENGELLIBITTAR
	ESKIHUKUMLU
	HUKUMACIKLAMA
	HUKUMKATEGORI
	DEVAMETTIGITAHSILI
	SIGARA
	SEYAHAT_EDEBILIR
	MESAI_KALABILIR
	TOPISTECRUBEYIL
	RESIM
	AKTIF
	ISYERI
	SICILNO
	MESLEGI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL

	ASKERLIKBASLANGICTARIHI
	ADAYID
	SIGORTA_ILK_GIR_TAR
	EMEKLILIK_TARIHI
	ASKERLIKTERHISACIKLAMA
	AKADEMIKUNVAN
	BITIRDIGITAHSILI
	BTAHSILID
	SOZLESMESEKLI
	SOZLESMEBITTAR
CVILAN	ISLETME_KODU
	ILANID
	ILANKODU
	ILANGRUBU
	ILANTIPI
	ALTBOLGE_KOD
	BOLUM_KOD
	DEPT_KOD
	POZISYON_KOD

	KADEME_KOD
	ULKEID
	IL_KOD
	GENELNITELIKLER
	ISTANIM
	PERSONELSAYISI
	CINSIYET
	ENGELLIILANIMI
	AKTIF
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
ISLETMELER	ISLETME_KODU
	ADI
CVEHLIYET	CVID
	EHLIYETTIP
	BELGENO
	VERILDIGIIL
	VERILDIGIILCE
	VERILISTARIHI

	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
EHLIYET	OZLUKID
	EHLIYETTIP
	BELGENO
	VERILDIGIIL
	VERILDIGIILCE
	VERILISTARIHI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	ID
ILETISIM	OZLUKID
	ISMAIL
	SAHSIMAIL
	DAHILI1
	DAHILI2
	ISCEPTEL

	SAHSICEPTEL
	EVTEL
	EVADRESI
	EVULKE
	EVIL
	EVILCE
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
CVILANYAYIN	INCKEYNO
	ILANID
	YAYINORG
	YAYINYERI
	YAYINBASTARIHI
	YAYINBITISTARIHI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	PERSONELSAYISI

NITELIK	ISLETME_KODU
	NITELIKID
	KISAACIKLAMA
	UZUNACIKLAMA
	GRUP
	TURU
	SURETIPI
	SURE
	KACUZERINDEN
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
ULKE	ULKEID
	ULKE_ISIM
	TELKODU
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
ORGGECMIS	ORGGECMISID

	OZLUKID
	BASTAR
	BITTAR
	NEDENI
	ALTBOLGE_KOD
	ORGULKEID
	ORGIL_KOD
	ORGILCEID
	BOLUM_KOD
	DEPT_KOD
	POZISYON_KOD
	GOREV_TANIMI
	AMIRID
	AMIRLINE
	MAVIBEYAZYAKA
	ISYERI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	ORGMASTERID

	KADEME_KOD
	ISLETME_KODU
DEPARTMAN	ISLETME_KODU
	DEPT_KOD
	DEPT_ISIM
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	BOLUM_KOD
	DEPT_ID
BOLUM	ISLETME_KODU
	BOLUM_KOD
	BOLUM_ISIM
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	ALTBOLGEKOD
KULLANICI	ISLETME_KODU
	KULLANICI_NO

	KULLANICI_ISIM
	AD
	SIFRE
	EMAIL
	AKTIVASYONKODU
	AKTIVASYONDURUMU
	KAYITTARIHI
	DUZELTMETARIHI
	SOYAD
	TWITTER_ID
	LINKEDIN_ID
	FACEBOOK_ID
	ENSON_SIFRETAR
	ENSON_LOGINTAR
	MERNISNO
	UYRUGU
	UYRUK_TIPI
	DOG_TARIH
CVTAHSIL	CVID
	ILKOKULISMI
	ILKMEZUN

	LISETIPI
	LISEBOLUMU
	LISEOKULISMI
	LISEBITISYILI
	LISEDILI
	LISEILI
	LISEMEZUN
	LISENOTSISTEMI
	LISEDERECE
	LISANS1TIPI
	LISANS1UNVTIPI
	LISANS1OGRTIPI
	LISANS1UNIVERSITE
	LISANS1FAKULTE
	LISANS1BOLUM
	LISANS1DILI
	LISANS1ILI
	LISANS1BURS
	LISANS1BASYIL
	LISANS1BITYIL
	LISANS1MEZUN

	LISANS1NOTSISTEMI
	LISANS1DERECE
	YUKSEKBOLUM
	YUKSEKILI
	YUKSEKBASYIL
	YUKSEKBITYIL
	YUKSEKMEZUN
	YUKSEKNOTSISTEMI
	YUKSEKDERECE
	DOKTORAUNVTIPI
	DOKTORAUNIVERSITE
	DOKTORABOLUM
	DOKTORAILI
	DOKTORABITYIL
	DOKTORAMEZUN
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
	LISANS1ACIKLAMA
	YUKSEKACIKLAMA

	DOKTORAACIKLAMA
	LISEHAZIRLIKEGITIMI
	LISANS1HAZIRLIKEGITIMI
	YUKSEKHAZIRLIKEGITIMI
	DOKTORAHAZIRLIKEGITIMI
	YUKSEKTEZ
	DOKTORAENSTITU
	YUKSEKENSTITU
	LISEULKESI
	LISANS1ULKESI
	YUKSEKULKESI
	DOKTORAULKESI
	YAYINLAR
OBJETIPI	OBJETIPKODU
	OBJEACIKLAMASI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
YETKINLIK	YETKINLIKID
	NITELIKID

	DEGERBIT
	DEGERI
	KISAACIKLAMA
	UZUNACIKLAMA
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
CVILAN_IL	ILANID
	IL_KOD
ALTBOLGE	ISLETME_KODU
	BOLGE_KOD
	BOLGE_ISIM
	ULKEID
	IL_KOD
	ILCEID
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
CVILANGRUP	ILANGRUPKOD

	ACIKLAMA
	KAYITYAPANKUL
	KAYITTARIHI
	DUZELTMEYAPANKUL
	DUZELTMETARIHI
POZISYONPROFILTASARIMSONUCLAR	ID
	OZLUKID
	ADISOYADI
	ISLETME_KODU
	DURUM
	TASARIMID
	TOPLAMAGIRLIK
	SONUCLAR
	SONHESAPLAMATARIHI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
FAKULTE	FAKULTEID
	FAKULTE_ISIM
	KAYITTARIHI

	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
POZISYONPROFILTASARIM	ID
	KODU
	ADI
	TOPLAMAGIRLIK
	TASARIM
	SONHESAPLAMATARIHI
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
POZISYONYETKINLIK	POZISYONYETKINLIKID
	POZISYON_KOD
	YETKINLIKID
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL
KADEME	ISLETME_KODU

	KADEME_KOD
	SAYISALDEGERI
	ACIKLAMA
	KAYITTARIHI
	KAYITYAPANKUL
	DUZELTMETARIHI
	DUZELTMEYAPANKUL

APPENDIX 3 – Primary Key list of tables

Primary keys of each table are shown. The key point of this table, a primary can have one or more than one column. Before creating the ontology, it must be considered.

Table Name	Primary key	Number of Columns
SICILGECMIS	TBLSICILGECMIS_PKEY	5
KULLANPROGP	TBLKULLANPROGP_PKEY	4
KULLANHAKP	TBLKULLANHAKP_PKEY	3
KULLANP	TBLKULLANP_PKEY	2
SICIL	TBLSICIL_PK	2
CVEHLIYET	TBLCVEHLIYETPK	2
CVILAN	CVILANPK	1
CVILANGRUP	PK__TBLCVILA__219682B93B47BAFE	1
CVILANYAYIN	CVILANYAYINPPK	1
CVTAHSIL	CVTAHSILPKEY	1
DEPARTMAN	DEPARTMANPKEY	1
EHLIYET	TBLEHLIYETPK	1
FAKULTE	FAKULTEPKEY	1
FIRMA	PK_TBLSIRKET	1
IL	ILPKEY	1

ILCE	ILCEPKEY	1
ILETISIM	ILETISIMPKEY	1
ISLETMELER	ISLETMELER_PKEY	1
KADEME	KADEMEPKEY	1
NITELIK	NITELIKPK	1
OBJETIPI	PK_OBJETIPI	1
ORGGECMIS	ORGGECMISPK	1
OZLUK	OZLUKPK	1
POZISYON	POZISYONPKEY	1
POZISYONPROFILTASARIM	POZPROFILTASARIMPKEY	1
POZISYONPROFILTASARIMSONUCLAR	POZPROFILTASARIMSONUCLARPKEY	1
POZISYONYETKINLIK	POZISYONYETKINLIKPK	1
PROFIL	PROFILPK	1
KULLANICI	TBLKULLANICI_PKEY	1
ALTBOLGE	ALTBOLGEPKEY	1
AUTHORIZATIONGROUP	PK_TBLAUTHORIZATIONGROUP	1
AUTHORIZATIONGROUPOBJECT	PK_TBLAUTHORIZATIONGROUPOBJECT	1
AUTHORIZATIONGROUPUSER	PK_TBLAUTHORIZATIONGROUPUSER	1

BOLUM	BOLUMPKEY	1
CV	CVPK	1
CVADAY	CVADAYPK	1
TAHSIL	TAHSILPKEY	1
ULKE	ULKELERPKEY	1
YETKINLIK	YETKINLIKPK	1

APPENDIX 4 – Foreign Key list of tables

#	FK_NAME	table	column	referenced_table	referenced_column
1	ALTBOLGEILFK	ALTBOLGE	IL_KOD	IL	IL_KOD
2	ALTBOLGEULKEFK	ALTBOLGE	ULKEID	ULKE	ULKEID
3	ALTBOLGEILCEFK	ALTBOLGE	ILCEID	ILCE	ILCEID
4	ALTBOLGEISLETMEFK	ALTBOLGE	ISLETME_KODU	ISLETMELER	ISLETME_KODU
5	FK_TBLAUTHGROUPUSER_TBLKULLANP	AUTHORIZATIONGROUPUSER	SIRKETKODU	KULLANP	ISYERI
6	FK_TBLAUTHGROUPUSER_TBLKULLANP	AUTHORIZATIONGROUPUSER	KULLANICINO	KULLANP	KULLANICI_NO
7	FK_TBLAUTHGROUPUSER_TBLAUTHORIZATIONGROUP	AUTHORIZATIONGROUPUSER	AUTHGROUPID	AUTHORIZATIONGROUP	ID
8	BOLUMALTBOLGEFK	BOLUM	ALTBOLGEKOD	ALTBOLGE	BOLGE_KOD

9	BOLUMISLETMEFK	BOLUM	ISLETME_KODU	ISLETMELER	ISLETME_KODU
10	CVADAYFK	CV	ADAYID	CVADAY	ADAYID
11	CVEHLIYETCVFK	CVEHLIYET	CVID	CV	CVID
12	CVEHLIYETILFK	CVEHLIYET	VERILDIGIIL	IL	IL_KOD
13	CVEHLIYETILCEFK	CVEHLIYET	VERILDIGIILCE	ILCE	ILCEID
14	CVILANALTBOLGEFK	CVILAN	ALTBOLGE_KOD	ALTBOLGE	BOLGE_KOD
15	CVILANDEPTFK	CVILAN	DEPT_KOD	DEPARTMAN	DEPT_KOD
16	CVILANBOLUMFK	CVILAN	BOLUM_KOD	BOLUM	BOLUM_KOD
17	CVILANILFK	CVILAN	IL_KOD	IL	IL_KOD
18	CVILANULKEFK	CVILAN	ULKEID	ULKE	ULKEID
19	CVILANPOZSFK	CVILAN	POZISYON_KOD	POZISYON	POZISYON_KOD

20	CVILANIL_IL_FK	CVILAN_IL	ILANID	CVILAN	ILANID
21	CVILANIL_CVILAN_FK	CVILAN_IL	IL_KOD	IL	IL_KOD
22	CVILANYAYINVFK	CVILANYAYIN	ILANID	CVILAN	ILANID
23	CVTAHSILCVFK	CVTAHSIL	CVID	CV	CVID
24	DEPARTMANBOLUMFK	DEPARTMAN	BOLUM_KOD	BOLUM	BOLUM_KOD
25	DEPARTMANISLETMEFK	DEPARTMAN	ISLETME_KODU	ISLETMELER	ISLETME_KODU
26	TBLEHLIYETILFK	EHLIYET	VERILDIGIIL	IL	IL_KOD
27	TBLEHLIYETILCEFK	EHLIYET	VERILDIGIILCE	ILCE	ILCEID
28	TBLEHLIYETOZLUKFK	EHLIYET	OZLUKID	OZLUK	OZLUKID
29	ULKEILFK	IL	ULKEID	ULKE	ULKEID
30	ILCEILFK	ILCE	IL_KOD	IL	IL_KOD
31	OZLUKEVILFK	ILETISIM	EVIL	IL	IL_KOD

32	OZLUK_EVULKE	ILETISIM	EVULKE	ULKE	ULKEID
33	OZLUKEVILCEFK	ILETISIM	EVILCE	ILCE	ILCEID
34	ILETISIMOZLUKFK	ILETISIM	OZLUKID	OZLUK	OZLUKID
35	KADEMEISLETMEFK	KADEME	ISLETME_KODU	ISLETMELER	ISLETME_KODU
36	TBLKULLANHAKP_FKEY1	KULLANHAKP	KULLANICI_NO	KULLANP	KULLANICI_NO
37	TBLKULLANHAKP_FKEY1	KULLANHAKP	ISYERI	KULLANP	ISYERI
38	TBLKULLANICIISLETMEFK	KULLANICI	ISLETME_KODU	ISLETMELER	ISLETME_KODU
39	TLBKULLANP_FK1	KULLANP	ISYERI	FIRMA	ISYERI
40	KULLANOZLUKFK	KULLANP	OZLUKID	OZLUK	OZLUKID
41	TBLKULLANPROGP_FKEY1	KULLANPROGP	ISYERI	KULLANHAKP	ISYERI
42	TBLKULLANPROGP_FKEY1	KULLANPROGP	KULLANICI_NO	KULLANHAKP	KULLANICI_NO

43	TBLKULLANPROGP_FKEY1	KULLANPROGP	MODUL_NO	KULLANHAKP	MODUL_NO
44	NITELIKISLETMEFK	NITELIK	ISLETME_KODU	ISLETMELER	ISLETME_KODU
45	ORGGECMISPOZSFK	ORGGECMIS	POZISYON_KOD	POZISYON	POZISYON_KOD
46	ORGGECMISKADEMEFK	ORGGECMIS	KADEME_KOD	KADEME	KADEME_KOD
47	ORGGECMISOZLUKFK	ORGGECMIS	OZLUKID	OZLUK	OZLUKID
48	ORGGECMISORGILCEFK	ORGGECMIS	ORGILCEID	ILCE	ILCEID
49	ORGGECMISALTBOLGEFK	ORGGECMIS	ALTBOLGE_KOD	ALTBOLGE	BOLGE_KOD
50	ORGGECMISBOLUMFK	ORGGECMIS	BOLUM_KOD	BOLUM	BOLUM_KOD
51	ORGGECMISORGILFK	ORGGECMIS	ORGIL_KOD	IL	IL_KOD
52	ORGGECMISDEPTFK	ORGGECMIS	DEPT_KOD	DEPARTMAN	DEPT_KOD
53	ORGGECMISORGULKEFK	ORGGECMIS	ORGULKEID	ULKE	ULKEID

54	OZLUKORGULKEFK	OZLUK	ORGULKEID	ULKE	ULKEID
55	OZLUKDEPTFK	OZLUK	DEPT_KOD	DEPARTMAN	DEPT_KOD
56	OZLUKORGILFK	OZLUK	ORGIL_KOD	IL	IL_KOD
57	OZLUKBOLUMFK	OZLUK	BOLUM_KOD	BOLUM	BOLUM_KOD
58	OZLUKORGILCEFK	OZLUK	ORGILCEID	ILCE	ILCEID
59	OZLUKALTBOLGEFK	OZLUK	ALTBOLGE_KOD	ALTBOLGE	BOLGE_KOD
60	OZLUKPOZSFK	OZLUK	POZISYON_KOD	POZISYON	POZISYON_KOD
61	OZLUKISLETMEFK	OZLUK	ISLETME_KODU	ISLETMELER	ISLETME_KODU
62	POZISYONISLETMEFK	POZISYON	ISLETME_KODU	ISLETMELER	ISLETME_KODU
63	POZISYONKADEMEFK	POZISYON	KADEMESI	KADEME	KADEME_KOD

64	POZISYONYETKINLIKPOZKODFK	POZISYONYETKINLIK	POZISYON_KOD	POZISYON	POZISYON_KOD
65	POZISYONYETKINLIKFK	POZISYONYETKINLIK	YETKINLIKID	YETKINLIK	YETKINLIKID
66	PROFILYETKINLIKFK	PROFIL	YETKINLIKID	YETKINLIK	YETKINLIKID
67	PROFILOZLUKFK	PROFIL	OZLUKID	OZLUK	OZLUKID
68	SICILOZLUKFK	SICIL	OZLUKID	OZLUK	OZLUKID
69	TBLSICIL_FKEY1	SICIL	ISYERI	FIRMA	ISYERI
70	OZLUKLISNS1FAKFK	TAHSIL	LISANS1FAKULTE	FAKULTE	FAKULTEID
71	YETKINLIKNITELIKFK	YETKINLIK	NITELIKID	NITELIK	NITELIKID