DOI: 10.22067/cke.2020.39289

Proposing an Integrated Multi Source Ontology Construction Methodology*

Research Article

Milanifard[1] Kahani[2]

Abstract: The goal of this research is to create a reference data model for educational and research institutes of Iranian Ministry of Sciences, Research, and Technology. After investigating existing technologies and considering the problem context, ontology was chosen as the data model format. In order to create the target ontology, an ontology construction methodology was designed and implemented. This methodology is created using design science research method and contains an architecture, a detailed workflow process, a guideline for performing 1each step, and related softwares in an integrated web-based system. The designed system is implemented in PHP and is available as open source. The system is used as the main tool to construct the target ontology. The proposed methodology leverages the three main knowledge sources including textual documents, existing ontologies in the higher education domain, and reverse engineering of a relational database of an integrated university system. The resulted product of this methodology was evaluated based on the data requirements of the Ministry of Sciences, Research, and Technology, and its shortcomings were resolved. The novelty of this work is both on the generated product, that is, a localized reference data model, and an ontology construction methodology.

Keywords: Ontology Development; Higher Education Ontology; Ontology Learning.

1. Introduction

This research was conducted according to the request of the Ministry of Science, Research and Technology (MSRT) of Iran for creating a reference ontology tailored to the educational and research domain in the higher education business.

Ontology is "a formal explicit specification of a shared conceptualization" [1] and is explained in a machine-readable language. In Information Technology, ontology is considered an information artifact that models a specific domain knowledge [2] and consists of classes (the representations of the real-world concepts), hierarchical relations between classes, data properties (expressing class attributes), and object properties (non-hierarchical relations between classes).

Ontologies can be constructed by using three types of knowledge resources: unstructured (such as text documents), semi-structured (such as HTML files) and structured (such as relational databases) resources [3]. MSRT required us to cover at least the following list of knowledge sources:

1. Statistical concepts of science, research and technology that are mentioned in two main books published by MSRT.
2. Data objects that are stored in an active higher education Enterprise Resource Planning (ERP) software in Iran;
3. All existing ontologies in this domain;

Ontology construction is an expensive and tedious task and must be done in a systematic way by applying a proper methodology.

Previous researches on creating higher education ontology since 2010 are listed in Table 1 and for each research, its methodology, resources, tools and the product is identified.

Table 1. The summary of previous work

Research	Methodology	Resources	Tools	Product
Satyamurty, Murthy, & Raghava [4]	Unknown	Unknown	Protégé	-
Hadjar [5]	Adopted from Enterprise Ontology [6]	Some universities organizational charts and executives of Ahlia University	Protégé	-
Zemmouchi-Ghomari & Ghomari [7]	Adopted from Neon [8]	Text documents and some web sites	Neon	HERO
Ameen, Khan, & Rani [9]	Proposed 7 methods without details	Unknown	Protégé	-
Malik, Prakash, & Rizvi [10]	Unknown	Unknown	-	-

As shown in Table 1, only two detailed methodologies have been used in previous work, that is, Enterprise ontology and Neon.

In order to find a suitable methodology for our project we further searched for other popular ontology construction methodologies. To compare the search result, we used a framework adapted from [11] that focuses on activity categories in the construction process (1. Management, 2. Pre-Development, 3. Development, 4. Post-Development, and 5. Support) and added our special criteria: supporting multi-language (persian in specific), having technical tools, and having detailed guidelines and algorithems (to support at least unstructured and structured knowledge sources). The result of this comparison is presented in Table 2.

* Manuscript received December, 4, 2019; accepted., 25 , .2020.

[1] Corresponding Author: Iranian Research Institute for Information Science and Technology (IRANDOC), Tehran, Iran.
Email: omid@um.ac.ir
[2] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.

Table 2. Ontology development methodologies comparison

Methodology	Supported activity category					Multi-Language support	Tools	Detailed
	1	2	3	4	5			
Enterprise Ontology [6]	-	-	X	-	X	-	-	No
METHONTOLOGY [12]	X	-	X	X	X	-	X	Partly
TOVE [13]	-	-	X	-	X	-	-	No
Ontology Development 101 [14]	-	-	X	-	X	-	-	No
DILIGENT [15]	X	-	-	X	X	-	-	No
UPON [16]	-	-	X	X	X	-	-	No
On-To-Knowledge [17]	X	X	X	X	X	-	X	No
Neon [8]	X	X	X	X	X	Localization	X	Partly

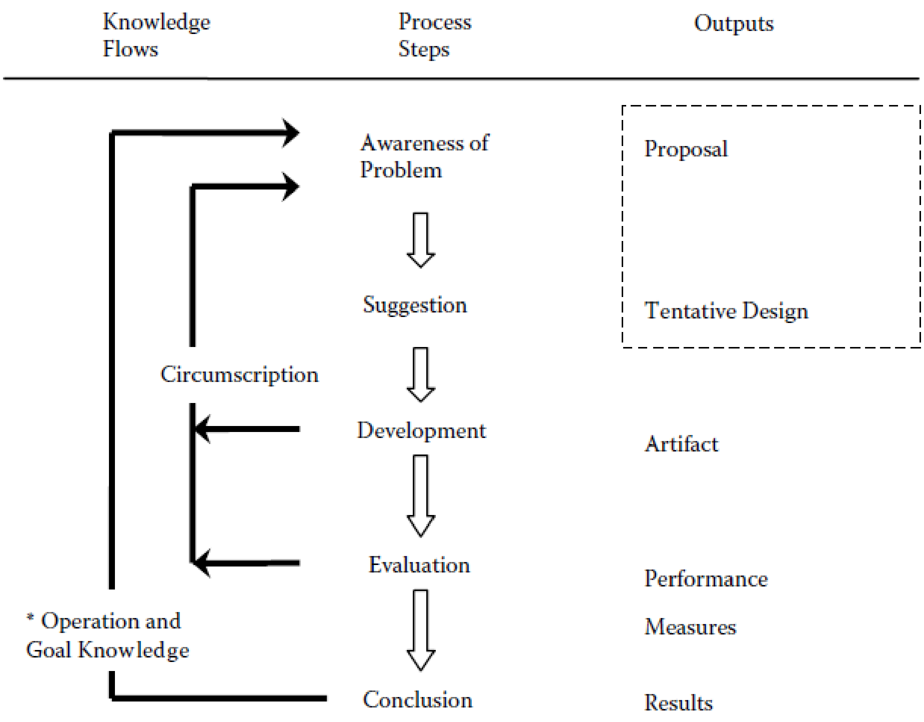

Figure 1. The general methodology of design science research [18]

As shown in Table 2, there is no comprehensive methodology that fits all criteria, thus we designed a new methodology by using Design Science Research (DSR) general cycle method, as shown in Figure 1, and then used this methodology to create the final product.

In "awareness of problem" step in DSR, we investigated different methods to construct ontology from three different types of knowledge sources: ontology learning from texts, ontology learning from relational databases, and creating an ontology by merging existing ontologies.

***Ontology learning[1] from texts*:**

Researchers suggested several semi-automated methods for learning ontology from texts. These methods can be categorized in three approaches: linguistic, statistical, and mixed [19].

In order to learn ontology from text, we selected a mix of TF-IDF[2] [20] and co-occurance analaysis [21] techniques from the statatistical approach, and Wordnet technique from the linguistic approach.

Ontology learning from Relational Databases:

Different methods and techniques were proposed by researchers for extracting ontology from a relational database. These methods can be categories in two main approaches: creating ontology based on database schemas, and domain-specific ontology [22]. The first approach relies on database schema and do not consider table contents and other meta-data such as existing vocabularies. The second approach considers database content and also knowledge of domain experts. In this work we focused on the second approach.

The domain-specific approach is also categorized into two sub-approaches: No-Reverse engineering and reverse engineering [22]. In the No-Reverse engineering approach, an RDF graph of database content is created and mapped to an ontology by experts (mostly manually). This approach is not suitable for large databases because the graph will be too large to create and investigate. In this research, our knowledge source is a higher education ERP database that contains more than 2000 tables, so we focused on re-

[1] Semi-automated ontology construction is also called ontology learning.
[2] Term Frequency (TF) – Inverse Document Frequency (IDF)

engineering the database approach.

Re-engineering methods use rules for transferring database entities to ontology elements. The following are the most used transfer rules [22]:

1. Default rules: These rules are adapted from Berners-Lee rules [23]: Briefly, these rules are: transfer tables to classes, non-foreign key fields to data properties, foreign key fields to object properties, and table records to instances.
2. Binary relationship rule: This rule identifies tables that are designed to link two tables and transfer them to object properties.
3. Hierarchy class rule: If the primary key of a table is a foreign key to the primary key of another table, there is a subclass-superclass relation between their mapped classes.
4. Weak entities rule: If a table has a composite primary key that contains a foreign key to another table, the mapped classes has a “part-of” relationship.
5. N-ary relationship rule: If a primary key consists of foreign keys to more than one table, it should be broken into binary relationships.
6. Fragmentation rule: If some tables have a same primary key, they should be integrated into one class.
7. Constraint rule: These rules exploit additional schema constraints, which are presented in SQL DDL statements (such as non-nullable and unique contraints)
8. Datatype rule: Transfer SQL datatype to value contraints in ontology.

The above rules are created based on some assumptions such as the database is in the 3NF format, tables and fields have meaningful names ,and all foreign keys are defined in the database schema. These assumptions are not true specially in large databases, so researchers suggested applying a three step process for extracting an ontology from a database [24]: preparation, extraction, and enrichment.

In the preparation step, we focus on two aspects of database elements meta-data:

1. Completeness: It means having a complete understanding and proper meta-data about database elements. All database entities should be labeled by meaningfull description and all relations between tables even those hidden in application code should be specified;
2. Relevence: Relevence of all database entities to our domain should be specified.

Table 3 shows the compaison of some highly cited re-engineering methods based on supporting preparation aspecs, extraction rules, and enrichment step. To the best of our knowledge there is no comprehensive re-engineering method that is suitable for our goal and thus we planned to design a new method that supports all mapping rules and the three steps of extraction.

***Creating an ontology by merging existing ontologies*:**
Existing ontologies are structured knowledge resources for ontology creation. Researchers have proposed several methods for ontology merging. These methods use at least one of the following approaches [39]:

1. Structure based: In this approach ontologies are represented as directed labelled graphs and similarity comparison between a pair of classes from two ontologies is based on the analysis of their position within the graphs. One of the popular methods in this approach is PROMPT [40].
2. Terminological based: Terminological methods compare strings and can be applied to the name, the label, or comments of ontology entities.
3. Instance based: These methods determine the similarity between concepts by examining the overlap of their instances.
4. Background knowledge based: Only few methods consider the background knowledge in the mapping process and they are limited to use knowledge in the upper ontology [41], knowledge hidden in corpus [42], and semantic web [43].

Most of tools and techniques for ontology merging, were developed as a part of a research project and were customized based on their needs [44]; therfore they beome outdated after elapsing a period of time. For example, PROMPT used to be a pioneer tool in ontology merging, however, it has not been updated in the past 10 years and the current version of protégé does not support it any more. Moreover, there is no tool or technique in ontology merging that supports Persian language especially in semantic similarity search by using Wordnet or other methods. Due to these reasons we decided to design a new method that supports structured and terminological based approachs and also use background knowledge, where the results of ontology learning from text and extracting ontology from database are the background knowledge.

Table 3. A comparison of re-engineering methods

Method	Preparation		Extraction rules								Enrichment
	Completeness	Relevance	1	2	3	4	5	6	7	8	
Shen, Huang, Zhu, & Zhao [25]			X	X	X			X	X	X	
Ghawi & Cullot [26]			X	X	X				X	X	
Tirmizi, Sequeda, & Miranker [27]			X	X	X		X		X	X	
Cerbah [28]		manually define relevance of database entities	X	X	X						X
Alalwan, Zedan, & Siewe [29]			X	X	X		X	X	X	X	
Lubyte & Tessaris [30]			X	X			X		X		
Albarrak & Sibley [31]			X	X	X				X	X	
Astrova [32]			X	X	X	X	X		X	X	X

Method	Preparation		Extraction rules								Enrichment
	Completeness	Relevance	1	2	3	4	5	6	7	8	
Liu, Wang, Bao, & Wang [33]			X	X			X		X		
Santoso, Haw, & Abdul-Mehdi [34]	identiying hirarchical relation based on table contents		X	X	X			X	X	X	
Khan & Sonia [35]			X	X	X	X	X		X		
Blobel [36]			X	X				X			
Kaulins & Borisov [37]			X	X	X	X	X	X	X	X	X
Zarembo [38]			X	X	X				X	X	

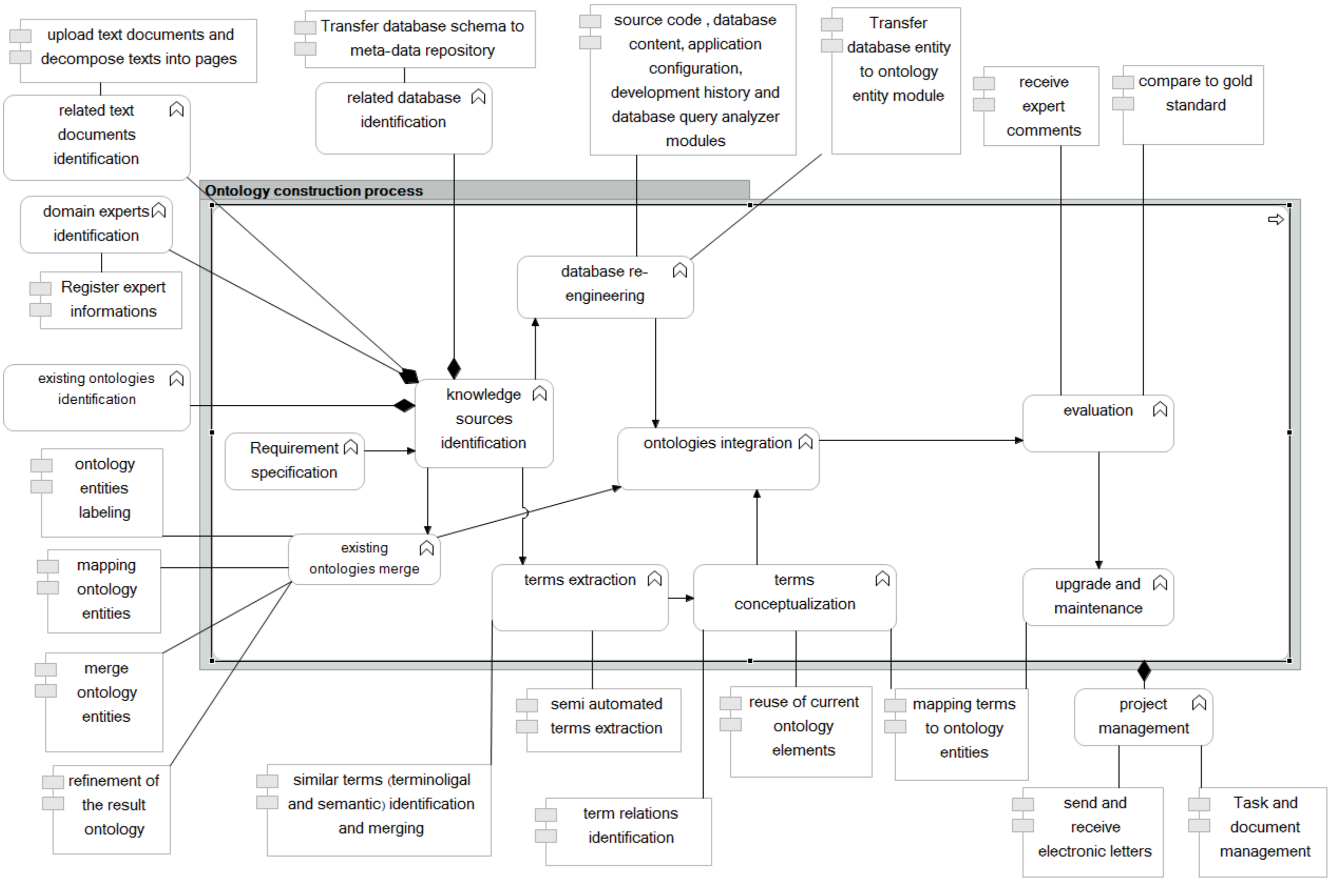

Figure 2. The ontology construction integrated system

2. Ontirandoc, an Integrated Methodology for Ontology Construction

The second phase of DSR is suggestion. In this phase we designed a tentative model of an integrated system, called Ontirandoc, which can be used for ontology construction from three types of knowledge sources. Ontirandoc is not only a tool for creating and editing ontology files, but also a methodology that contains detailed process guideline, methods, algorithms, and an integrated modular software to support the process[1].

In the development phase of DSR, we implemented our algorithms in an open source PHP web application. The implemented system was tested by input data (text documents, ERP database and existing ontologies) and the results were checked manually to find exceptions and errors. The system was evolved according to the results of the evaluation phase.

After passing several rounds of “suggestion – development – evaluation” cycle in DSR, we reached our final integrated system. This system has a modular design and is open source to enable other researchers to upgrade or customize it according to their especial needs.

The structure of the system is shown in Figure 2. The model was designed by ArchiMate[2] language that is one of the architecture description languages in ISO/IEC/IEEE 4210.

The proposed structure covers main activities for ontology construction and also provides a platform for collaborative ontology development. The main activities were adapted from Methontology [12], On-To-Knowledge [17] and Neon [8] methodologies.

Several software modules were designed and implemented in an integrated system to support main activities and Persian language. The modularity design

[1] According to [45] definition, methodology is “a comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought intensive work ought to be performed”

[2] http://pubs.opengroup.org/architecture/archimate3-doc/

allows to upgrade or customize each module independently. All modules are integrated based on data layer as shown in Figure 3.

Ontirandoc activities and related modules:

1. Requirement specification: Almost all ontology construction methodologies consider this activity for which the result is a document that specifies the goal, scope, and requirements of the product.
2. Knowledge sources identification: environment study and feasibility test are two main tasks that are mentioned in On-To-Knowledge and Neon methodologies.

In Ontirandoc, these tasks are decomposed into four activities: existing ontologies identification, domain expert identification, related text documents identification, and related database identification.

To identify existing ontologies, we designed a 6 steps guideline as following. Step 1-4 are adapted from ontology dowsing document suggested by [46]:

- Checking list of ontologies and services websites
- Using semantic search engines (such as swoogle[1])
- Checking ontology repositories
- Checking mailing lists and online forums.

We extended ontology dowsing guideline by adding 2 steps:

- When an ontology is found, investigate its code and if it uses other ontology elements, find and check the referred ontologies.
- Search scientific articles that may have an ontology result.

1. Terms extraction: In this activity, ontology developers extract terms based on open coding technique in content analysis methods [47]. The first time a term is identified by a developer, he can add it and its location (page, paragraph and sentence) into the terms vocabulary by using Ontirandoc register terms user interface. The location will be used in co-occurance analysis. If developers identify an existing term in the text, they can select it from vocabulary and add its new location, so the system can calculate TF-IDF of each term. Some modules are designed and implemented in order to help developers to:

- Identify previously extracted terms.
- Suggest similar existing terms before adding a new one. This module will show both structural and semantic similarity. Semantic similarity is identified by using wordnet (in our case we used a Persian wordnet called FerdowsNet[2]) and structure similarity is identified by Levenshtein distance and prefix/suffix analysis.
- Merge similar terms.

2. Terms conceptualization: The goal of this activity is transfering terms to ontology entities. Developers may create a new ontology entity for a term or just map the term to an existing ontology elements. A software module calculates TF-IDF value of extracted terms and shows them as a sorted list to developer. A term with larger TF-IDF is more important in that domain. The following software modules help developers in this activity:

- Showing a term references in texts. By selecting each term, this module shows all paragraphs that have this term.
- Showing semantic related terms (in the current version just synonyms, hyponyms and hypernyms) for each selected term by using WordNet and FerdowsNet. These lists would help developers to identify hierarchical or non-hierarchical relations in the ontology.

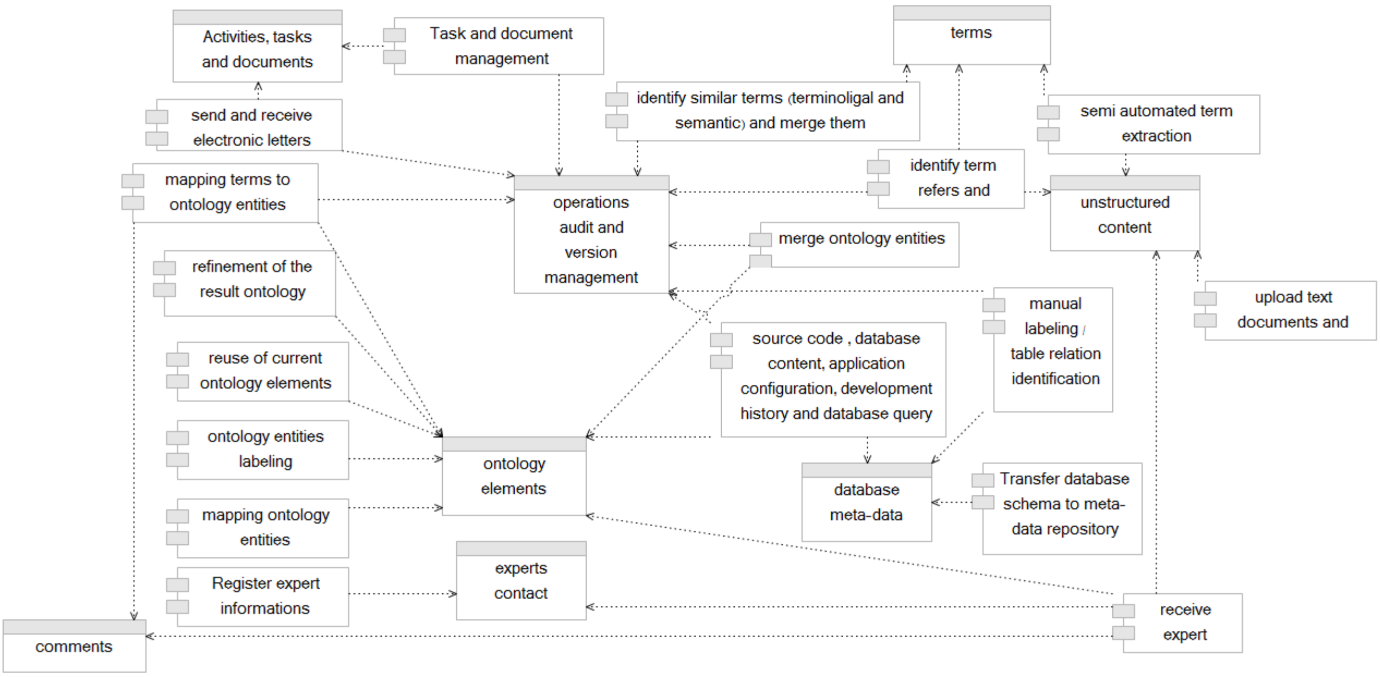

Fig. **1** Integration and relations between modules in data layer

1 http://swoogle.umbc.edu
2 http://wtlab.um.ac.ir/index.php?option=com_content&view=article&id=314&Itemid=200

- Showing similar terms (structural similarity) for each selected term. This list helps developers to identify relations between classes or properties of classes.
- Showing similar ontology elements in existing ontologies. It assists developers to select a better ontology element type by knowing other's modeling view.
- Performing co-occurrence analysis to identify relation between terms and their mapped ontology elements.

After conceptualizing all the terms, following software modules would help developers to refine the result ontology:

- Showing all classes that have similar child classes and asking the developer if he wants to merge them.
- Showing redundant properties/relations (exists in both parent and child class) and asking the developer if he wants to remove them.
- Showing similar relations between two classes and asking the developer if he wants to merge them.

1. Database re-engineering: This activity is designed in two steps: preparation and extraction.

Ontirandoc relies on a rich meta-data, therefore the preparation step is designed to prepare such data. A rich meta-data should have the following information about database elements:

- All elements should have clear and meaningful labels that describe their content and existence reason. These labels can be defined in Persian and English languages.
- Relatedness of each element to business domains should be specified.
- All table relations should be specified (some of these relations are defined in the database schema and some of them are hidden in applications' code).

To support enrichment of meta-data, several modules were designed and implemented in Ontirandoc:

- Table content investigator: Researchers have proposed a few solutions to extract the meaning of tables by analyzing their contents, such as [34] and [48]); however these solutions are not efficient for large tables like the case of our database. The table content investigator module in Ontirandoc does not apply any specific data mining or other data processing algorithms and only allows ontology developers to investigate table contents by applying horizontal and vertical filters.
- Source code investigator: Most of the ambiguities in database entities meaning can be resolved by investigating application source code [24]. Some table relations might also be hidden in the source code. This module proposes a practical solution to complete the meta-data by investigating application source code. Ontology developers can use this module through a user interface that allows them to complete meta-data of a table through following features:
    - Showing all source files that send queries with specific table names to the DBMS[1] (it assumes that this module has access to query log files). Ontology developers can trace usage of a table in source files and identify the meaning of that table by reading the related source codes.
    - Showing content of a source code file to the developer.
    - Showing source code files evolution history (it assumes that this module has access to the software project management data). History of a source file helps ontology developers to find the reasons of creation and evolution of a source code that is related to a table. It also helps to discover software developers who work on that source file, and may need to refer to software developers and ask them about the usage of a table.
    - Investigating the software configuration: information systems usually organize their features in system menus. Relation between software menus and source code files is a good knowledge source about the meaning of tables. This module helps ontology developers to trace a menu from the source files that use specific tables. Description of menus can tell ontology developers about the meaning of tables and also ontology developers can refer to those menus in functional systems and extract the meaning from their UI[2].
    - Suggesting table relations: The structural similarity between a field name in one table and primary key in another table may reveal a foreign key that is not defined in the database schema.

In the extraction step of re-engineering, several algorithms were designed to implement 8 transferring rules that we discussed before. These algorithms relay on a complete meta-data that were prepared in the preparation step.

Figure 4 shows the algorithm of applying default, weak entities, and constraint rules (rules number 1, 5, and 7). Key ideas of this algorithm are considering the coding tables and restricted values of fields.

As presented in Figure 4, each non-key field is transferred to a data property, because in large databases, like our case, the result has too many data properties. In this case, the prepared meta-data is very helpful. Ontirandoc extraction module adds all similar data properties in a list. Two data properties are similar if their title or label (in Persian) are structurally or semantically similar. Moreover, if two data properties have the same permitted values list, they might also be similar. Ontology developers can review the list and select which data properties should be merge together.

Figure 5 shows the algorithm of applying binary and N-

[1] Database Management System
[2] User Interface

ary relationship rules (rules number 2 and 5).

Figure 6 shows applying hierarchy and fragmentation rules algorithm (rules number 3 and 6). This algorithm identifies potential fragmented tables and hierarchy relations according to the meta-data and allows user to confirm or reject the suggestions.

2. Merging existing ontologies: This activity has four steps: labeling ontology elements, mapping similar ontology elements, merging ontologies, and refining the result. Four software modules were designed in correspondence to these steps.

Labeling step will provide localized (each element has a Persian label) and consistent (all same elements have same label) ontologies. Ontirandoc software modules and UI help

ontology developers to navigate between ontologies and their elements, view structural and semantically similar elements, and add proper labels.

Because of the difference in naming and modeling view, finding similar elements in different ontologies cannot be fully automated and needs user intervention [49]. Figure 7 shows Ontirandoc suggested workflow for performing this step.

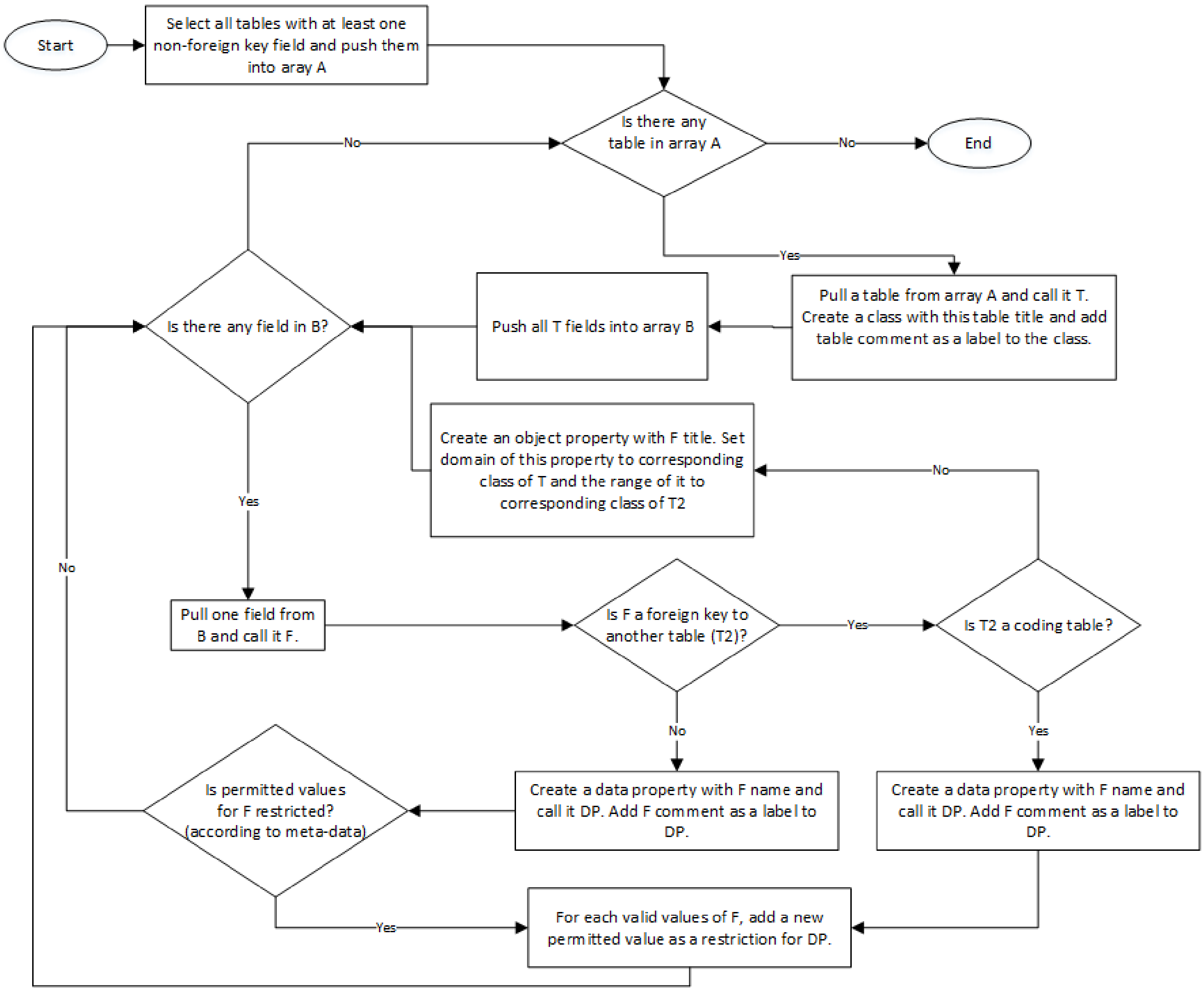

Figure 4. Algorithm for rules number 1, 5, and 7

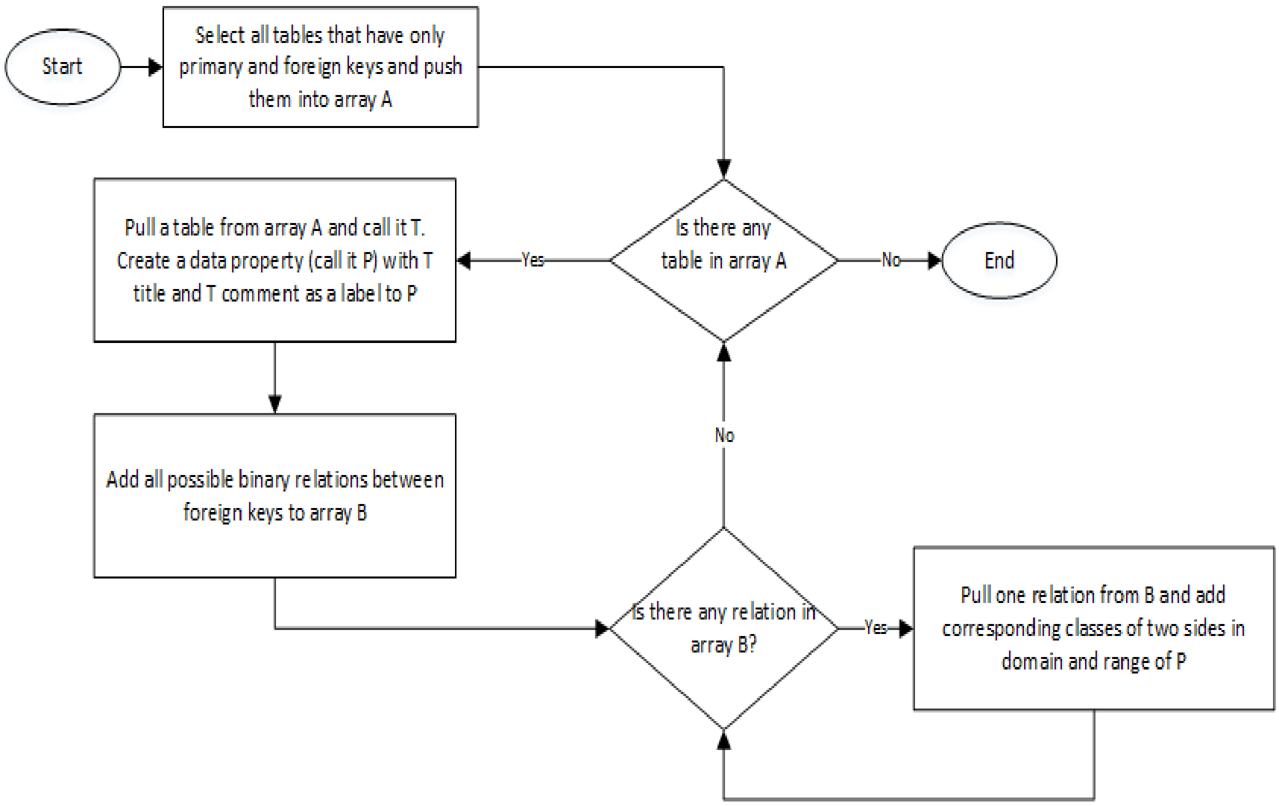

Figure 5. Algorithm for rules number 2 and 3

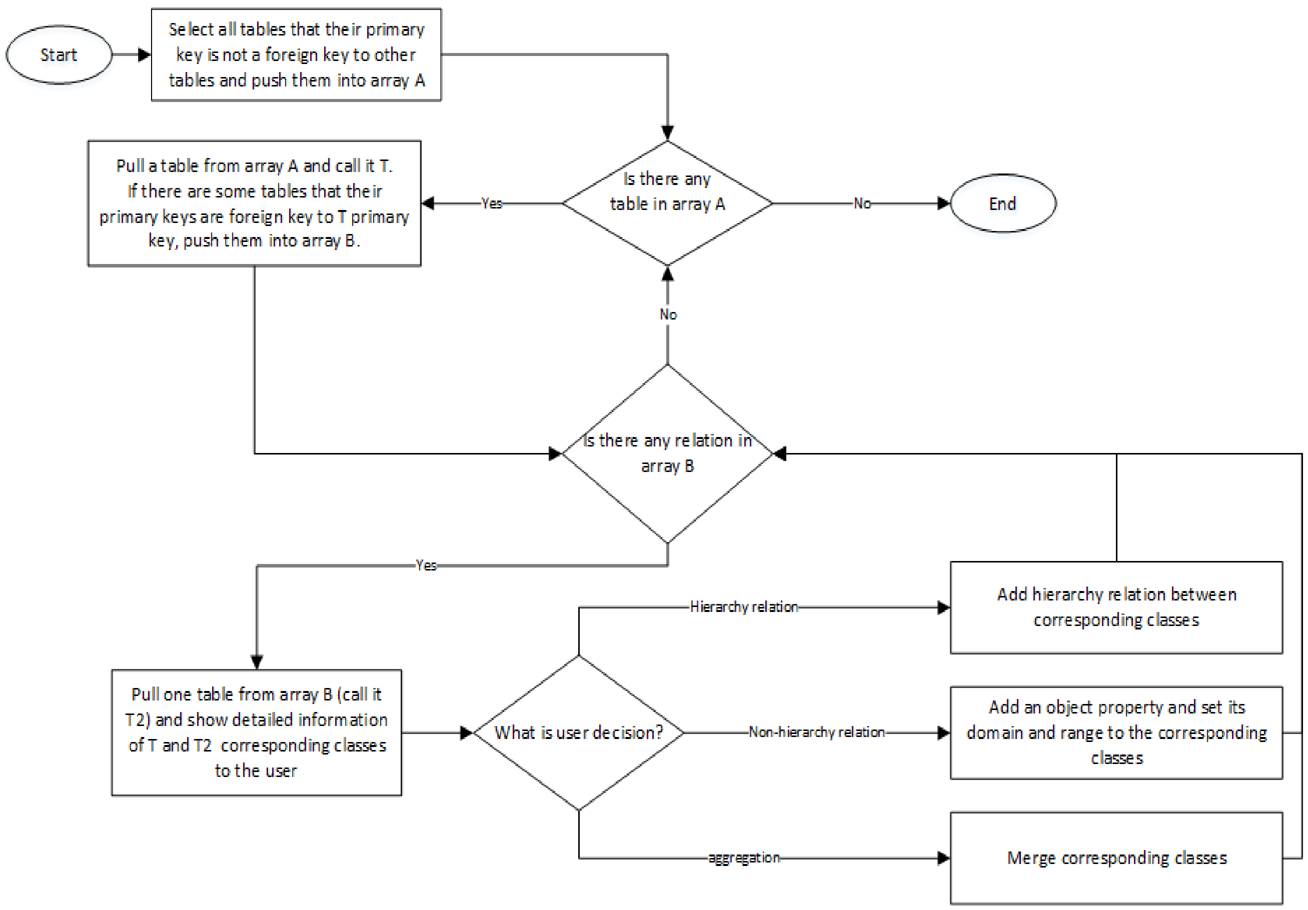

Figure 6. Algorithm for rules number 3 and 6

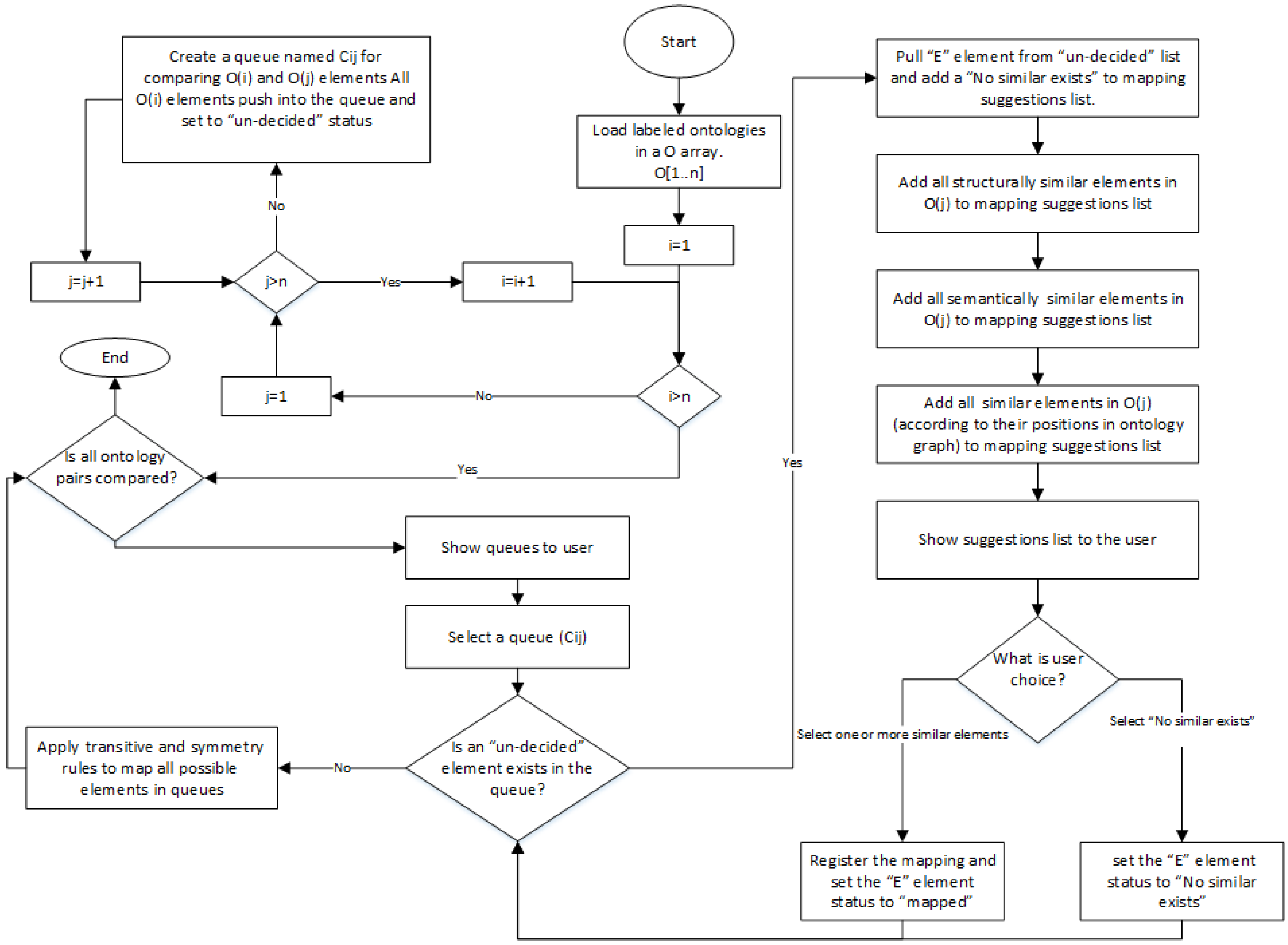

Figure 7. Mapping process workflow

The merge step also needs the user intervention. Figure 8 shows the Ontirandoc suggested algorithm for merging ontologies based on the results of previous step. Having enough documentation about ontology elements is a very important issue in application of an ontology [7]. The merge algorithm like other designed methods in Ontirandoc allows users to track each ontology element to its source.

The last step in merging activity is refinement. Because of the difference in granularity, detail level, and modeling view of source ontologies, the product of previous step may have some errors. Ontirandoc methodology suggests the following operations in the refinement step (these operations can be performed by the software modules that are designed and implemented in Ontirandoc):

- Identifying and investigating similar relations: If two classes have more than one semantic relation, these relations may be duplicate. These classes should be shown to the user in order to merge or remove redundancy.
- Identifying duplicate properties: Classes with hierarchy relations should not be in domain or range of a property. Because of the inheritance between parent and child classes, these duplications should be found and fixed.
- Suggesting hierarchy relations: Classes that their common properties and relations are more than a threshold, may have hierarchy relation. These classes should be shown to the user, so that he can select one of the following choices:
  - Selecting one class as parent and removing all common properties and relations from the child classes;
  - Creating a new class as parent of all selected classes; Removing all common properties and relations from child classes and inserting them into the new class;
  - Do nothing;

3. Evaluation: some researchers have proposed several methods to evaluate an ontology. These methods can be classified into three approaches [50]: comparing ontology with a "golden standard" based on the user, based on application of ontology, and based on comparing with the source of data. In our methodology, the evaluation activity is designed based on two approaches:

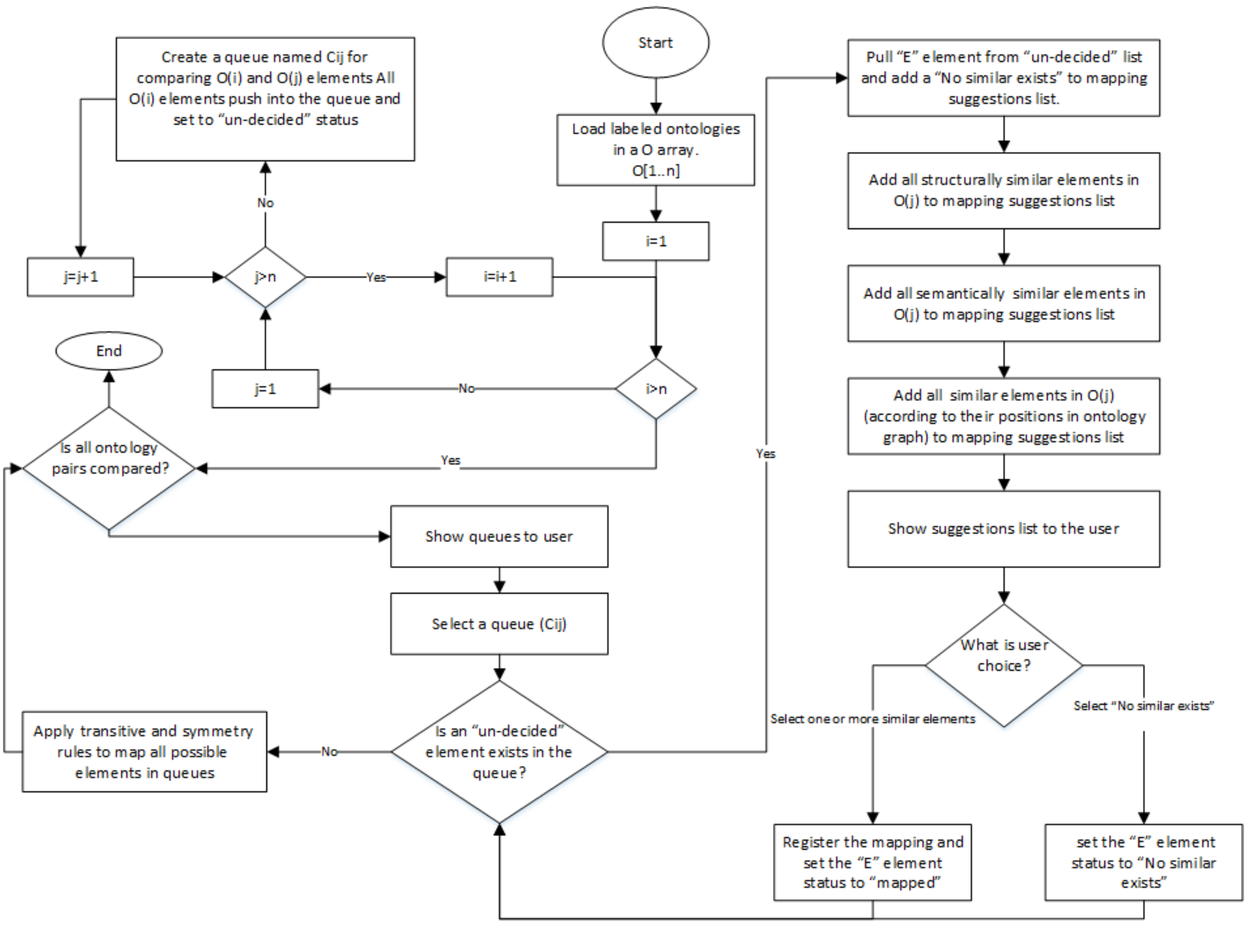

Figure 8. Merge algorithm

Table 4. Ontologies created by Ontirandoc

Ontology resource	Number of classes	Number of properties/relations
Existing ontologies	135	165
Text	83	172
Database	156	655

- Comparing with a golden standard: precision and recall are two main measures that should be calculated [51]. A software module was designed and implemented to calculate these parameters. It is worth noting that before comparing two ontologies, their elements must be labeled by using Ontirandoc tools as we discussed before.
- Based on user: Assertions technique is one of the methods in this approach. This would allow users to investigate data model details by viewing them in a list of natural language assertions [52]. We adapted this technique, customized it to support Persian language, and implemented a web-based software module to show an ontology details in Persian language assertions and get users opinion and comments. The user's feedback is aggregated and shown to developers for updating the ontology.

In addition to checking validity of ontology by applying the above approaches, we designed and implemented a software module to calculate the quality of the target ontology based on the framework presented in [53]. Ontology quality measures that are implemented in these modules are Number of Properties (NOP), Average Properties per Class (AP-C), Average Fanout of Classes (AF-C), Number of Roots (NoR), and Average Fanout of Root Classes (AF-R).

***3. Constructing the Target Ontology*:**

We used Ontirandoc methodology to construct our target ontology. In the knowledge source identification activity, the following sources are identified:

1. Existing ontologies: 8 related ontology OWL files on the web are identified by using the upgraded ontology dowsing method:

- Common European Research Information Format (CERIF)[1]
- Lehigh University Benchmark (LUMB) [2]

[1] http://www.eurocris.org/Uploads/Web%20pages/CERIF-1.6/CERIF_1.6_2.xsd
[2] http://swat.cse.lehigh.edu/onto/univ-bench.owl

- Semantic Web for Research Communities (SWRC)[1]
- Toronto University[2]
- University Ontology[3]
- VIVO[4]
- National Current Research Information System for IRAN (SEMAT) [54]
- Higher Education Reference Ontology (HERO)[5]

2. Text documents: "Statistical concepts of science, research and technology" [55] and "Statistics of Higher Education in Iran (Academic Year 2015-2016)" [56] books both published by Higher Education Research and Planning institue.
3. Database: Ferdowsi University of Mashhad ERP database.

After performing all activities before final ontologies integration, we obtained three products from three different knowledge sources that are shown in Table 4.

As you can see in Table 5, comparing these ontologies with each other shows that none of them fully covers other concepts and properties. The first number in each cell shows number of classes in the row ontology that have corresponding classes in the column ontology, and the second number shows number of properties in the row ontology that have corresponding properties in the column ontology.

The ontology that is constructed from database has the most details (properties and relations). This is because of the nature of ERP database that should contain almost all operational data structure in a specific domain, but it does not cover about 30% of concepts and properties of the two other ontologies. Some of these concepts are not designed in the database because their corresponding business process is not automated, such as "Audit Board", and others are super classes that are designed in more than one table, such as "Publication". The goal of Ontirandoc is constructing a comprehensive ontology as much as possible, so the final activity is performed to integrate these three ontologies into the final ontology with 164 classes and 585 data and object properties (OWL file of this ontology can be downloaded from GitHub[6]).

Table 5. Comparing overlap of ontologies on each other

	From existing ontologies	From text	From database
From existing ontologies		61.45% 27.91%	72.19% 71.51%
From text	41.95% 31.52%		72.79% 65.45%
From database	41.03% 20.31%	47.44% 18.47%	

There exists no golden standard ontology for higher education in Iran, therefore we requested some colleagues to create a new ontology based on MSRT information gathering systems[7]. We assumed that these systems cover almost all data needed by MSRT, so it may be used as a benchmark to calculate the recall parameter of ontology evaluation. This benchmark ontology is created by a simple manual re-engineering method, that is, investigating user interface forms and transferring forms and their elements to ontology elements.

The created benchmark ontology has 55 classes, and 155 objects and data properties[8].

The comparison between the final ontology and this benchmark shows that the final ontology has a big difference in covering the benchmark ontology elements compared to existing ontologies. As shown in the second column of Table 6, it covers almost all elements of the benchmark ontology. Moreover, three experts used the implemented user-based evaluation method that was mentioned earlier and their comments show that the final ontology is valid.

Table 6 shows the quality measures of the final ontology compared to existing ontologies that presents its high quality.

Table 6. Comparing the final ontology with other ontologies

Ontology	Coverage of benchmark (recall measure)	AF-R	NoR	AF-C	AP-C	NoP	NoC
VIVO	41.43%	50765	1	132.55	0.66	252	383
CERIF	36.19%	85.5	2	0.83	3.77	781	207
SEMAT	35.71%	16359	2	207.08	1.22	193	158
HERO	25.24%	1769	2	63.18	2.52	141	56
LUMB	11.43%	40	2	2	0.68	27	40
SWRC	18.57%	704	1	13.28	1.06	56	53
Toronto Ontology	13.33%	96	2	3.76	0.65	33	51
University Ontology	16.67%	21.6	5	1.57	0.62	43	69
Final ontology	96.19%	21744	1	132.59	6.35	1041	164

[1] http://swrc.ontoware.org/ontology
[2] http://www.cs.toronto.edu/semanticweb/maponto/MapontoExamples/univ-cs.owl
[3] http://www.webkursi.lv/luweb05fall/resources/university.owl
[4] http://vivoweb.org/files/vivo-isf-public-1.6.owl
[5] http://sourceforge.net/projects/heronto/
[6] https://github.com/milanifard/HigherEducationOntology
[7] Higher Education System (http://hes.msrt.ir), SAHMA (https://portal.irphe.ac.ir) and SEMAT (http://www.semat.ir)
[8] https://github.com/milanifard/HigherEducationOntology

4. Conclusion

In this research we created a reference ontology for education and research domain of higher education in Iran. This ontology was constructed by a new methodology that was designed using DSR method and contains an architecture, detailed workflow process, and guideline for performing each step. In order to implement and test this methodology (according to DSR life cycle), we developed an integrated modular open source web-based system that supports all activities mentioned in our methodology.

The designed system was implemented in over 40,000 lines of code in PHP. It can be download from GitHub[1] and it is free to use, customize and add new modules to suuport special needs of other researchs and projects.

A reference ontology for education and research organizations of Ministry of Science, Research and Rechnology was built using Ontirandoc methodology and its integrated system. This product was validated by experts and was also compared with MSRT information needs (benchmark ontology). The quality measures show the final product has a high quality.

References

[1] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907-928, 1995

[2] N. Guarino, D. Oberle, and S. Staab, “What is ontology?” In Handbook, *Handbook on ontologies*. Springer, Heidelberg, Berlin, pp. 1-17, 2009.

[3] A. Zouaq, D. Gasevic and M. Hatala, “Towards open ontology learning and filtering,” *Information Systems*, vol. 36, no. 7, pp. 1064-1081, 2011.

[4] C. V. Satyamurty, J. V. Murthy and M. Raghava, “Developing Higher Education Ontology Using Protégé Tool: Reasoning,” *In Smart Computing and Informatics,”* Springer , Singapore, pp. 233-241, 2018.

[5] K. Hadjar, “University Ontology: A Case Study at Ahlia University,” *In Semantic Web,* Springer, Cham, pp. 173-183, 2016.

[6] M. Uschold and M. Gruninger, “Ontologies: Principles, methods and applications,” *The Knowledge Engineering Review*, vol. 11, no. 2, pp. 93-136, 1996.

[7] L. Zemmouchi-Ghomari and A. R. Ghomari, “Process of Building Reference Ontology for Higher Education,” *In Proceedings of the World Congress on Engineering vol. 3*, WCE, London, U.K., pp. 1595-1600, 2013.

[8] M. C. Suarez-Figueroa, “NeOn methodology for building ontology networks: Specification, scheduling and reuse (Doctoral dissertation),” Univ. Politécnica, Madrid, 2010.

[9] A. Ameen, K. R. Khan and B. P. Rani. “Construction of university ontology,” *In Information and Communication Technologies (WICT), 2012 World Congress*, IEEE, Trivandrum, India, pp. 39-44, 2012.

[10] S. K. Malik, N. Prakash and S. A. Rizvi, “Developing an university ontology in education domain using protégé for semantic web,” *International Journal of Science and Technology,* vol. 2, no. 9, pp. 4673-4681, 2010.

[11] A. Gomez-Perez, M. Fernández-López and O. Corcho, “Ontological engineering: With examples from the areas of Knowledge Management, e-Commerce and the Semantic Web,” Springer Science & Business Media, 2006.

[12] M. Fernández-López, A. Gómez-Pérez and N. Juristo, “Methontology: from ontological art towards ontological engineering”, Univ. Politécnica, Madrid, 1997.

[13] M. Grüninger and M. S. Fox, “Methodology for the design and evaluation of ontologies”. in Proc. of *the Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI-95 , Montreal, 1995.

[14] N. F. Noy and D. L. McGuinness, “Ontology development 101: A guide to creating your first ontology,” Stanford University, Stanford, CA, 2001.

[15] H. S. Pinto, C. Tempich and S. Staab, “Ontology engineering and evolution in a distributed world using DILIGENT,” In *Handbook on ontologies,* Springer-Verlag, Berlin, Heidelberg, pp. 153-176, 2009.

[16] A. De Nicola, M. Missikoff and R. Navigli, “A software engineering approach to ontology building”. *Information Systems*, vol. 34, no. 2, pp. 258-275, 2009.

[17] Y. Sure, S. Staab and R. Studer, “On-to-knowledge methodology (OTKM),” In *Handbook on ontologies*, Springer, Berlin, Heidelberg, pp. 117-132, 2004.

[18] V. K. Vaishnavi and W. Kuechler, “Design science research methods and patterns: innovating information and communication technology”. Crc Press, New York, 2015.

[19] A. Zouaq and R. Nkambou, “A survey of domain ontology engineering: methods and tools,” In *Advances in intelligent tutoring systems*, Springer, Berlin Heidelberg, pp. 103-119, 2010.

[20] C. Sammut and G. I. Webb, Encyclopedia of Machine Learning, Springer, Boston, MA, 2011.

[21] P. Cimiano, “Ontology Learning and Population from Text Algorithms: Evaluation and Applications,” Springer-Verlag, New York, USA, 2006.

[22] D. E. Spanos, P. Stavrou and N. Mitrou, “Bringing relational databases into the semantic web: A survey,” *Semantic Web,* vol. 3, no. 2, pp. 169-209, 2012.

[23] T. Berners-Lee, “Relational databases and the semantic web (in design issues),” *World Wide Web Consortium*, https://www.w3.org/DesignIssues/RDB-RDF.html, 1998.

[24] B. El Idrissi, S. Baïna and K. Baïna, “A methodology to prepare real-world and large databases to ontology learning,” *Enterprise Interoperability VI,* Springer, pp. 175-185, 2014

[1] https://github.com/milanifard/Ontirandoc

[25] G. Shen, Z. Huang, X. Zhu and X. Zhao, "Research on the Rules of Mapping from Relational Model to OWL," In *Proceedings of the OWLED'06 Workshop on OWL: Experiences and Directions*, Athens, GA, USA, 2006.

[26] R. Ghawi and N. Cullot, "Database-to-ontology mapping generation for semantic interoperability," *VDBL'07 conference, VLDB Endowment,* ACM, Vienna, Austria, pp. 1-8, 2007.

[27] S. H. Tirmizi, J. Sequeda and D. Miranker, "Translating sql applications to the semantic web," *International Conference on Database and Expert Systems Applications*, Springer, Berlin, Heidelberg, pp. 450-464, 2008.

[28] F. Cerbah, "Mining the content of relational databases to learn ontologies with deeper taxonomies," *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, IEEE Computer Society, Sydney, NSW, Australia, pp. 553-557, 2008.

[29] N. Alalwan, H. Zedan and F. Siewe, "Generating OWL ontology for database integration," *Advances in Semantic Processing, SEMAPRO'09. Third International Conference*. IEEE, Sliema, Malta, pp. 22-31, 2009.

[30] L. Lubyte and S. Tessaris, "Automatic extraction of ontologies wrapping relational data sources," *International Conference on Database and Expert Systems Applications,* Springer, Berlin, Heidelberg, pp. 128-142, 2009.

[31] K. M. Albarrak and E. H. Sibley, "Translating relational & object-relational database models into OWL models," *Information Reuse & Integration, 2009. IRI'09. IEEE International Conference*, Las Vegas, NV, USA, pp. 336-341, 2009.

[32] I. Astrova, "Rules for mapping SQL relational databases to OWL ontologies," In *Metadata and Semantics,* Springer, Boston, MA, pp. 415-424, 2009.

[33] P. Liu, X. Wang, A. Bao and X. Wang, "Ontology automatic constructing based on relational database," *Grid and Cooperative Computing (GCC), 2010 9th International Conference*, IEEE, Nanjing, China, pp. 412-415, 2010.

[34] H. A. Santoso, S. C. Haw and Z. T. Abdul-Mehdi, "Ontology extraction from relational database: Concept hierarchy as background knowledge," *Knowledge-Based Systems,* vol. 24, no. 3, pp. 457-464, 2011

[35] S. Khan, K. Sonia, "R2o: Relation to ontology transformation system," *Journal of Information & Knowledge Management,* vol. 10, no. 01, pp. 71-89, 2011.

[36] B. Blobel, "Conceptual Model Formalization in a Semantic Interoperability Service Framework: Transforming Relational Database Schemas to OWL," *11th International Conference on Wearable Micro and Nano Technologies for Personalized Health,* IOS Press , Vienna, Austria, p. 35, 2014.

[37] A. Kaulins and A. Borisov, "Building Ontology from Relational Database," *Information Technology and Management Science,* vol. 17, no. 1, pp. 45-49, 2014.

[38] I. Zarembo, "Automatic Transformation of Relational Database Schema into OWL Ontologies," *International Scientific and Practical Conference (Vol. 3),* Rezekne, Latvia, pp. 217-222, 2015.

[39] K. Ramar and G. Gurunathan, "Technical review on ontology mapping techniques," *Asian Journal of Information Technology,* vol. 15, no. 4, pp. 676-688, 2016.

[40] N. F. Noy and M. A. Musen, "The PROMPT suite: interactive tools for ontology merging and mapping," *International Journal of Human-Computer Studies,* vol. 59, no. 6, pp. 983-1024, 2003.

[41] R. E. Menéndez-Mora and R. Ichise, "Ontology matching by actively propagating user feedbacks through upper ontologies," *Revista Vínculos,* vol. 10, no. 2, pp. 85-92, 2013.

[42] J. Madhavan, P. A. Bernstein, A. Doan and A. Halevy, "Corpus-based schema matching". *In Data Engineering, ICDE 2005 Proceedings. 21st International Conference*. IEEE, Tokyo, Japan, pp. 57-68, 2005.

[43] P. Jain, P. Hitzler, A. P. Sheth, K. Verma and P. Z. Yeh, "Ontology alignment for linked open data," *International Semantic Web Conference*. Springer, Berlin, Heidelberg, pp. 402-417, 2010.

[44] L. Otero-Cerdeira, F. J. Rodríguez-Martínez and A. Gómez-Rodríguez, "Ontology matching: A literature review," *Expert Systems with Applications,* vol. 42, no. 2, pp. 949-971, 2015.

[45] *IEEE Guide for Software Quality Assurance Planning. Std. 730.1–1995 IEEE Computer Society,* 1995.

[46] W3C., *Ontology Dowsing [online]*. *ESW Wiki*. Retrieved from http://esw.w3.org/Ontology Dowsing, 2010, January 14.

[47] J. Corbin and A. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Sage Publications, 2008

[48] D. Octaviani, A. Pranolo and S. Othman, "RDB2Onto: an approach for creating semantic metadata from relational educational data," *Science in Information Technology (ICSITech), 2015 International Conference*. IEEE, Yogyakarta, Indonesia, pp. 137-140, 2015.

[49] S. Massmann, S. Raunich, D. Aumüller, P. Arnold and E. Rahm, "Evolution of the COMA match system," *In Proceedings of the 6th International Conference on Ontology Matching-Volume 814,* CEUR-WS. Org, Bonn, Germany, pp. 49-60, 2011.

[50] L. Ouyang, B. Zou, M. Qu and C. Zhang, "A method of ontology evaluation based on coverage, cohesion and coupling," *In Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on (Vol. 4).* IEEE, Shanghai, China, pp. 2451-2455, 2011.

[51] J. Brank, M. Grobelnik and D. Mladenić, "A survey of ontology evaluation techniques," *In Proceedings of the*

Conference on Data Mining and Data Warehouses (SiKDD) , Ljubljana, Slovenia, 2005.

[52] G. Simsion and G. Witt, Data modeling essentials, Morgan Kaufmann, San Fransisco, 2005.

[53] A. M. Orme, H. Yao and L. H. Etzkorn, “Indicating ontology data quality, stability, and completeness throughout ontology evolution,” *Journal of Software: Evolution and Process*, vol. 19, no. 1, pp. 49-75, 2007.

[54] M. J. Khoshroo and O. Fatemi, "SEMAT, National Current Research Information System for IRAN," In *Connecting Science with Society: the Role of Research Information in a knowledge-based Society. 10th International Conference on Current Research Information Systems (CRIS2010)*, Aalbort, Denmark, pp. 27-33, 2010.

[55] A. Hamzelooi, E. Khodabakhsh, H. Javdani, A. Davari, D. Hatami, M. Shirazi, R. Norooz zadeh, Statistical concepts of science, research and technology. Higher Education Research and Planning institue, Tehran, 2015.

[56] A. Saeedi, Statistics of Higher Education in Iran (Academic Year 2015-2016). Higher Education Research and Planning institue, Tehran, 2016.