Márcio Louzada de Freitas

Goal and Risk Analysis in the Development of Information Systems for the Web of Data

Vitória, ES

2018

Márcio Louzada de Freitas

Goal and Risk Analysis in the Development of Information Systems for the Web of Data

Master's Dissertation presented to the Postgraduate Program in Computer Science of the Federal University of Espírito Santo, as partial requirement to obtain the Degree of Master in Computer Science.

Federal University of Espírito Santo

Faculty of Technology

Postgraduate Program in Computer Science

Supervisor: Prof. Vítor E. Silva Souza, Ph.D.

Vitória, ES

2018

Márcio Louzada de Freitas

Goal and Risk Analysis in the Development of Information Systems for the Web of Data/ Márcio Louzada de Freitas. – Vitória, ES, 2018-

94 p. : il. (algumas color.) ; 30 cm.

Supervisor: Prof. Vítor E. Silva Souza, Ph.D.

Mestrado em Informática – Federal University of Espírito Santo
Faculty of Technology
Postgraduate Program in Computer Science, 2018.

1. Goal Modeling. 2. Risk Modeling. 3. Linked Data. 4. Vocabularies. I. Souza, Vítor Estêvão Silva. II. Universidade Federal do Espírito Santo. III. Goal and Risk Analysis in the Development of Information Systems for the Web of Data

CDU 02:141:005.7

Márcio Louzada de Freitas

Goal and Risk Analysis in the Development of Information Systems for the Web of Data

Master's Dissertation presented to the Post-graduate Program in Computer Science of the Federal University of Espírito Santo, as partial requirement to obtain the Degree of Master in Computer Science.

Trabalho aprovado. Vitória, ES, XX de dezembro de 2018:

Prof. Vítor E. Silva Souza, Ph.D.
Orientador

João Henrique C. Pimentel
Universidade Federal Rural de Pernambuco

Monalessa Perini Barcellos
Universidade Federal do Espírito Santo

Vitória, ES
2018

Acknowledgements

Thank God for giving me strength to do this work and for leading my life. My family for helping me get here. To my supervisor and professor Vítor Estêvão Silva Souza, for the opportunity to enter the master's degree, for the past teachings and orientation. The teacher Renata Guizzardi for the contribution given to the paper and this dissertation. The evaluation members of this dissertation for having accepted the invitation for participation and contribution. I also thank my fellow students and all those who contributed directly and indirectly to this work.

Abstract

Linked Data is a set of data published on the World Wide Web in an interconnected fashion, whose contents can be processed by machines, forming a Web of Data. Published data and their interconnections are described by means of vocabularies, i.e., schemas that describe the existing entities and the relationship between them. Moreover, such data can refer to several domains, such as Geographic, Media, Social Media, Governmental, Libraries and Education, Life Sciences and so on.

The publication of Linked Data on the Web leads to new problems related to Requirements Engineering, which needs to take into account aspects related to new ways of developing systems and delivering information. In this context, tasks such as functional and non-functional requirements elicitation and ontology-based conceptual modeling can be applied to the development of systems that publish Linked Data, in order to obtain a better shared conceptualization of the published data (i.e., a domain ontology).

The use of vocabularies is an intrinsic activity when publishing or consuming Linked Data and their choice can be supported by the elicited requirements and domain ontology. The use of GORE (Goal-Oriented Requirements Engineering) modeling languages, such as iStar, can be employed in requirements elicitation, as well as help identify actors, agents and roles, and to model their goals, tasks and resources, aiming at the development of information systems which are integrated with the Web of Data. Also, risk identification, modeling and analysis techniques can be employed, in order to identify risks and their impacts on stakeholder goals.

In this work, we propose GRALD: Goals and Risks Analysis for Linked Data, an approach for modeling goals and risks for information systems for the Web of Data. Our proposal aims to present tool support for the creation of goal and risk models, helps the process of choosing vocabulary through best practices and suggests a catalog of goals, risks, tasks and resources related to Linked Data.

GRALD combines two existing approaches. On the Web Engineering side, the FrameWeb-LD approach aids developers on publishing the data from Web-based Information Systems as Linked Data, connecting it to vocabularies that can be specified on UML-based diagrams during architectural design. On the Requirements Engineering side, the RISCOSS approach seeks to align business goals and risks in the adoption of Open Source systems. We adapted the latter so it can be applied to the adoption of Linked Data. GRALD seeks to establish a synergy between these two approaches.

Keywords: Goal Modeling, Risk Modeling, Linked Data, Vocabularies.

List of Figures

List of Tables

List of abbreviations and acronyms

DCAT	Data Catalog Vocabulary
GBRAM	Goal-Based Requirements Analysis Method
GORE	Goal-Oriented Requirements Engineering
GRALD	Goals and Risks Analysis for Linked Data
HTML	Hyper-Text Markup Language
IoT	Internet of Things
KAOS	Knowledge Acquisition in autOmated Specification
LD	Linked Data
LOV	Linked Open Vocabularies
NFR	Non-functional Requirements
OSS	Open Source Software
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RDFa	Resource Description Framework Attributes
RE	Requirements Engineering
SD	Strategic Dependence
SR	Strategic Rationale
UFO	Unified Foundational Ontology
URI	Uniform Resource Identifiers
XML	Extensible Markup Language

Contents

1 Introduction

This chapter presents the context, motivation, objectives, method and organization of this dissertation.

1.1 Context

The Semantic Web was presented by Berners-Lee, Hendler and Lassila (2001) as the Web version that seeks to make content understandable by both humans and machines, improve search engines by giving meaning to published content and take into account contextual information of time, space and states of things. According to its creators, a challenge of the Semantic Web is to ensure the expressiveness of the published content, without losing performance in the representation of the data on the Web. Ontologies are used to integrate different databases, to define the classes, subclasses and relationships between them for the creation of contents for the Semantic Web, making it possible to generate inferences (BERNERS-LEE; HENDLER; LASSILA, 2001).

At the core of the Semantic Web idea is the concept of *Linked Data*.[1] According to Bizer, Heath and Berners-Lee (2009), Linked Data is a set of data interconnected by URIs (Uniform Resource Identifiers)[2] whose contents can be processed by machines, forming a Web of Data. The published content is based on the RDF (Resource Description Framework) standard[3] and data can be extracted using SPARQL[4] queries. Also, according to the authors, vocabularies can be used to describe the entities and the relationship between them, and are expressed in RDF format.

There are several vocabularies for the representation of published content. Meta-data languages such as Microformats, Microdata and RDFa (Resource Description Framework Attributes) can be used to annotate Web pages with semantic content (POHOREC; ZORMAN; KOKOL, 2013).

According to Heath and Bizer (2011), data published as Linked Data refer to several domains, such as Geographic, Media, Social Media, Governmental, Libraries and Education, Life Sciences and so on. With the adoption and implementation of Linked Data in several areas of knowledge by companies, institutions and governments, it becomes necessary to analyze goals and requirements of stakeholders and systems, as well as to identify and analyze the risks of adopting Linked Data in Web-based Information Systems. Given that

1 <http://linkeddata.org/>
2 <https://www.w3.org/wiki/URI>
3 <https://www.w3.org/RDF/>
4 <https://www.w3.org/TR/rdf-sparql-query/>

the Web follows an open and decentralized architecture, connecting an information system with external data sources can lead to potential risks, thus the need to understand their impact on stakeholder goals.

Goal-Oriented Requirements Engineering (GORE) approaches aim to analyze the goals of systems and stakeholders not in an isolated way but in an integrated environment of more than one system, being able to take into account functional and non-functional requirements (LAPOUCHNIAN, 2005). GORE approaches, such as the NFR Framework (MYLOPOULOS; CHUNG; NIXON, 1992), iStar (YU, 2009) and KAOS (Knowledge Acquisition in autOmated Specification) (LAMSWEERDE; LETIER, 2000) could be applied to the modeling of Web-based Information Systems and, in particular, to analyze the use of Linked Data by such systems.

Some approaches combine goal modeling with risk modeling, which provide tools that help analyze the impact of risks on stakeholder goals. For instance, the GR Framework (ASNAR; GIORGINI; MYLOPOULOS, 2011) allows modeling and reasoning about risks during requirements analysis. KAOS, mentioned above, allows not only goal modeling but also obstacle analysis. The RISCOSS project (COSTAL et al., 2015) proposes to integrate risk modeling language RiskML with goal modeling language iStar to analyze risks in the adoption of open source software. With some effort, these approaches can be adapted to the analysis of risks in the development of Web-based Information Systems that publish Linked Data. Henceforth, we refer to such systems as *Linked Data Systems*.

1.2 Motivation

Motivated by the growing publication of Linked Data in various domains (ANALYTICS, 2019), there are several works related to the development of Linked Data Systems. Our particular focus in on the application of goal modeling and risks analysis to the development of such systems. The following points motivated us to apply such particular approach:

1. In Linked Data Systems development there are specific functional and non-functional requirements, tasks, and resources employed. Because the published data are interlinked, stakeholder interests and goals should be considered. Goals could assist in the requirements specification task, provide criteria for a complete analysis and to define which requirements are relevant; justify the importance of requirements to stakeholders; help to detect and eventually resolve conflicts between requirements, etc. (LAMSWEERDE, 2001);

2. In this context, specific risks related to Linked Data can be considered, for example, risks related to vocabulary adoption, data publication, creation and maintenance of

ontologies, among others. Risks related to traditional Web development can also be taken into account. Therefore, analyzing the impact of risk events on the goals of stakeholders has particular relevance;

3. Since vocabularies describe (i.e., give meaning to) the published data and their interconnections, choosing inappropriate vocabularies can lead to several problems, such as misinterpretation of meanings due to poor documentation, connection timeouts due to infrastructure problems, etc. Thus, developers need support when choosing and creating vocabularies for data publication, taking into account best practices related to vocabulary adoption.

To our knowledge, there are no approaches for modeling goals and for identification and modeling of risks related to Linked Data. Thus, we propose an approach to assist the developer in identifying and analyzing goals and risks on the development of Linked Data Systems, as well as choosing good vocabularies for the publication of Linked Data.

1.3 Objectives

The general objective of this work is to support developers of Web-based Information Systems in the analysis of goals and risks regarding the adoption of Linked Data and the choice of appropriate vocabularies. The main objective is decomposed in the following specific objectives:

1. Adopt an existing goal-oriented requirements language for modeling Linked Data System requirements;
2. Adopt an existing risk modeling language in order to support risk analysis with respect to the use of Linked Data, integrated with system requirements;
3. Develop tools for creating and maintaining goal and risk models built with the chosen languages;
4. Suggest a catalog of goals, risks, tasks and resources for Linked Data, in order to aid developers in the creation of goal and risk models;
5. Provide systematic techniques that can aid developers in the choice of good Linked Data vocabularies;
6. Evaluate the proposal on actual Linked Data Systems, which refer to different domains.

1.4 Method

In order to fulfill the objectives above, we performed the following activities:

1. **Literature Review**: analysis of relevant publications in the areas of Semantic Web, Linked Data, Goal-Oriented Requirements Engineering and Requirements-based Risk Analysis was conducted in order to properly identify the gap on which we would focus our efforts;

2. **Baseline Definition**: existing approaches on Risk Analysis and Linked Data Systems development were studied in order to compose the baseline of our approach. We decided to adopt the RISCOSS approach (COSTAL et al., 2015) (see also the project website[5]) which uses iStar (YU, 2009) to create goal models and RiskML language (COSTAL et al., 2015) to create risk models, together with the FrameWeb-LD approach (CELINO et al., 2016), which focuses on publication of Linked Data by Web-based Information Systems;

3. **Proposal**: after the literature review, we elaborated the GRALD approach, seeking synergy between RISCOSS and FrameWeb-LD, performing goal and risk modeling with iStar and RiskML (RISCOSS), respectively, for the publication of Linked Data with FrameWeb-LD. As a result of this research a paper was published (FREITAS et al., 2018);

4. **Tool Support**: once the approach was defined, we searched for tools to create goal and risk models and to aid in the search of Linked Data vocabularies. As a result of this research, we chose to use the piStar[6] Web tool to create goal models, which supports iStar 2.0, is easy to use and extend and is open source. We extended the piStar tool in order to create the risk model and integrated goal-and-risk model, based on the metamodel of RiskML. For the search of vocabularies we particularly used LOV (Linked Open Vocabularies),[7] due to its ease of use and quantity of data sets and existing vocabularies;

5. **Evaluation**: finally, our proposal were evaluated using Web-based Information Systems developed by students in the context of the *Web Development & Semantic Web* course of our Postgraduate Program in Computer Science, all of which aim to publish Linked Data. The same is available for external consultation, in a public source code repository.[8] The experience of applying GRALD to different Linked Data Systems also allowed us to propose a catalog of goals, risks, tasks and resources for Linked Data.

5 <http://www.riscoss.eu>.
6 <http://www.cin.ufpe.br/~jhcp/pistar/tool/>
7 <http://lov.okfn.org/>.
8 <https://github.com/nemo-ufes/FrameWeb-GRALD>

1.5 Organization

The first and current chapter of this dissertation is the **Introduction**, which presents the context, motivation, objectives, methodology and organization of this work. The remainder of the dissertation is divided as follows:

- **Chapter 2 – Literature Review**: summarizes the basic concepts on top of which we built our proposal. Concepts related to Semantic Web and Linked Data, GORE approaches and Risks in software development are presented;
- **Chapter 3 – Proposal**: presents the GRALD approach, detailing its process for goal modeling, risk modeling and vocabulary choice for Web-based Information Systems that publish Linked Data. Tools for creating and maintaining the models and a catalog of goals, risks, tasks and resources for Linked Data are introduced;
- **Chapter 4 – Evaluation**: reports on the evaluation of our proposal, in which the method is applied to Linked Data Systems in different domains in order to verify its applicability;
- **Chapter 5 – Conclusion**: concludes, discusses limitations and future work.

2 Literature Review

This chapter addresses basics concepts related to the Semantic Web (Section 2.1), Goal-Oriented Requirements Engineering (Section 2.2) and Risk Analysis (Section 2.4). It also presents the approaches that form the baseline of our work: the iStar goal modeling language (Section 2.3), the RISCOSS approach (Section 2.5), the FrameWeb-LD method (Section 2.6). Finally, Section 2.7 presents the conclusion of this chapter.

2.1 The Semantic Web

The first version of the Web was created in 1989. In (TEKLI et al., 2013), this version of the Web (*Web 1.0*), was known as the *"Web of Documents"*, in which pages are written in HTML (*Hyper-Text Markup Language*), uniquely identified by a URI and interlinked through hyperlinks. The second version, the *Social Web*, is characterized by a greater interaction between users, also actively involved in content publication.

The traditional Web has some limitations. The content published is aimed at human consumption, with concerns related to layout, colors, images, sizes, etc. Searches on the Web are made by keywords and navigation by links, and the execution of complex searches can be very difficult and sometimes do not bring the expected results (HORROCKS, 2007).

In this context, The Semantic Web was introduced by Berners-Lee, Hendler and Lassila (2001) as an extension of the current Web, which aims to give a better defined meaning to the content and enabling machines and people to work together. This scenario allows software agents to process the content in favor of the users.

According to Tekli et al. (2013), the Semantic Web, known as Web 3.0, aims to extend the Web by giving information a well-defined meaning, in order to improve data accessibility to humans and machines. The machines will be able to automatically explore the semantic content, in different places and distributed environment. Moreover, others contributions are the improvement of search engines and data accessibility; better data integration and presentation; and provide more intelligent services for users, due to the greater interaction between user and machine, services based on geolocation, etc.

Figure 1, shows technologies such as URI, RDF, XML (*Extensible Markup Language*), RDFS (*Resource Description Framework Schema*), SPARQL Query, OWL (*Ontology Web Language*), among others, which are employed in the Semantic Web, some of which will be addressed in the following subsections. Layers related to trust and proof of the information, are also taken into account. User Interface and Applications would use all this technology in order to fulfill the Semantic Web vision.

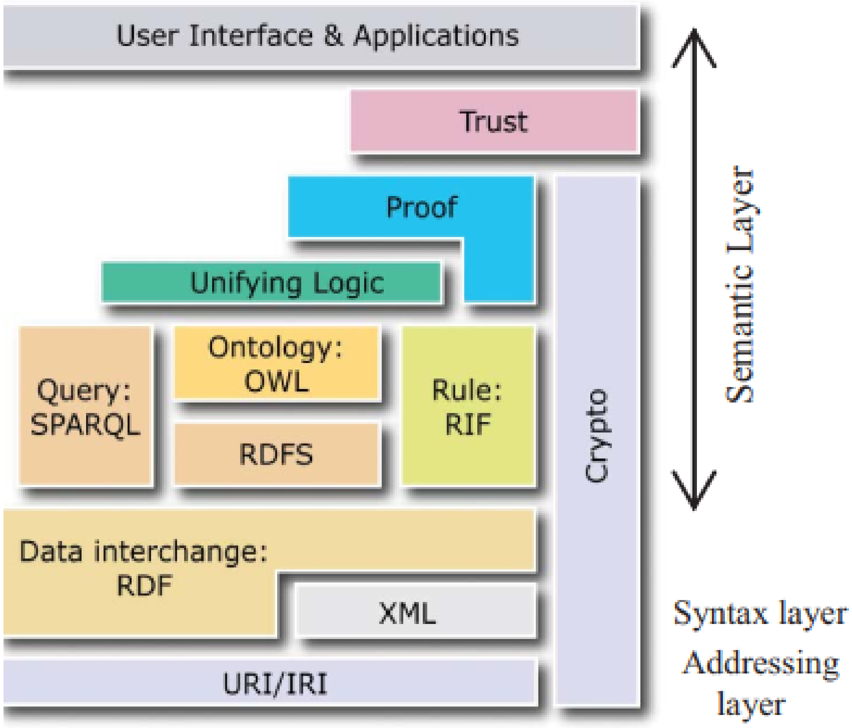

Figure 1 – The Semantic Web Stack (TEKLI et al., 2013).

The Semantic Web can benefit from other technologies related to IoT (Internet of Things), physical objects such as smartphones, smartcars, robotics systems, among others; autonomous software agents can be considered (TEKLI et al., 2013).

The World-Wide Web Consortium[1] (W3C) supports the Semantic Web. From the *"Web of Documents"* the W3C helps to build the technological stack for *"Web of Data"* or Linked Data, data from databases which are published on the web in an interconnected way, so that it can be processed by computers (W3C, 2018b).

2.1.1 RDF and RDFS

According to Tekli et al. (2013), while XML addresses the syntactic/structural properties of data, RDF is based on XML to better manage semantic interoperation. RDF is a data model designed to standardize the definition and use of metadata in order to better describe and deal with data semantics.

Pohorec, Zorman and Kokol (2013) explain that RDF has the purpose to represent graph-based information (entities of the world) on the Web of Data. An RDF triple consists of a subject, a predicate (the attribute of the triple) and an object. Meta-data languages such as Microformats, Microdata and RDFa can be used to annotate Web pages with semantic content.

According to Bizer, Heath and Berners-Lee (2009), the subject and object are

[1] https://www.w3.org/standards/semanticweb/

Subject: `http://dig.csail.mit.edu/data#DIG`
Predicate: `http://xmlns.com/foaf/0.1/member`
Object: `http://www.w3.org/People/Berners-Lee/card#i`

Subject: `http://data.linkedmdb.org/resource/film/77`
Predicate: `http://www.w3.org/2002/07/owl#sameAs`
Object: `http://dbpedia.org/resource/Pulp_Fiction_%28film%29`

Figure 2 – Example RDF links (BIZER; HEATH; BERNERS-LEE, 2009).

both URI's that each identify a resource, or a URI and a datatype (e.g., string) literal, respectively. The predicate is also represented by a URI, and specifies how the subject and object are related. Figure 2 shows two examples of RDF links. In the first, the object https://www.w3.org/People/Berners-Lee/card#i is member of the subject http://dig.csail.mit.edu/data#DIG, related by the predicate http://xmlns.com/foaf/0.1/-member. In the second example the subject http://data.linkedmdb.org/resource/film/77 is the same as http://dbpedia.org/resource/Pulp_Fiction_%28film%29, related by the predicate http://www.w3.org/2002/07/owl#sameAs, i.e., in the real world the subject and the object are the same movie.

According to the W3C (2014), mechanisms are provided for describing groups of related resources and their relationships. RDFS models and manipulates classes, just as it is done by object-oriented programming languages. RDFS uses concepts related to classes and properties such as rdfs:Class, rdfs:SubClassOf, rdfs:SubPropertyOf for inheritance; and predicate for the specification of domain and range of relations: rdfs:Domain and rdfs:Range (TEKLI et al., 2013).

2.1.2 Ontology

In the context of the Semantic Web, Linked Data is described by ontologies. Guarino, Oberle and Staab (2009) seeks to define ontology in two ways. In the first, as a philosophical discipline, that is, the branch of philosophy which deals with the nature and structure of reality. In the second case, it refers to an artifact, an ontology, which is a way of modeling a system, entities and relations between them.

Guarino (1998) classifies the types of ontology according to the different levels of generality. **Top-level ontologies** are independent of a particular domain or problem and describe general concepts such as space, time, material, object, event, action, among other. **Domain ontologies** describe the vocabulary related to a particular domain, such as Law, Medicine, Education, Requirements Engineering, etc. **Task ontologies** describe task or activity such as registration or sale of a product. Finally, **application ontologies** describe concepts that depend on a particular domain and task.

Guizzardi (2005) claims that ontologies can be applied in different areas of Computer and Information Sciences such as Information Systems, Domain Engineering, Artificial Intelligence and Semantic Web. It can also be used in domains such as Engineering and Technical Applications, Medicine, Law, among others. Ontologies are part of the stack of technologies involved in the Semantic Web. According to Berners-Lee, Hendler and Lassila (2001) the most typical kind of ontology for the Web has a taxonomy and a set of inference rules, in which the taxonomy defines classes of objects and relations among them.

Ontologies can also be classified in *reference* or *operational* ontologies. While **Reference Ontologies** are focused on human representation, using highly expressive languages, **Operational Ontologies** are focused on machine consumption and concerned about certain desired computational properties (GUIZZARDI, 2007). OWL is the W3C standard language for operational ontologies. The Semantic Web is largely based on operational ontologies written using the OWL language, presented next.

2.1.3 OWL

According to Horrocks (2008), RDF has some limitations, for example, it is not able to describe cardinality constraints or even a simple conjunction of classes. In this context, the need arose for a more expressive ontological language. In 2004, OWL was created, by a working group of W3C researchers.

OWL is a standard for describing ontologies on the Web (TEKLI et al., 2013). It extends RDF and RDFS, presenting constructs such as owl:Class, owl:ObjectProperty and owl:DataTypeProperty which extend rdfs:Class and rdfs:Property. OWL allows property specialization, identification of disjoint classes, specification of cardinality, etc. Listing 2.1 shows the class of Student Wizards written in RDF/XML format Horrocks (2008). This example shows how two existing classes, Student and Wizards, can be combined in an intersection to form a new class.

Listing 2.1 – Example of OWL in RDF/XML.

```
<owl:Class>
  <owl:intersectionOf
     rdf:parseType="Collection">
       <owl:Class rdf:about="#Student"/>
       <owl:Class rdf:about="#Wizard"/>
    </owl:intersectionOf>
</owl:Class>
```

In (W3C, 2012), the complete syntax of OWL 2 (Second Version) is presented, with axioms related to classes, such as EquivalentClasses, DisjointClasses, SubClassOf, DisjointUnion, IntersectionOf; and to properties, such as subObjectPropertyOf, equivalentObjectProperties, objectPropertyDomain; among other axioms and categories.

2.1.4 Vocabularies

Given an operational ontology language, such as OWL, vocabularies can be created to represent information published in the form of Linked Data. According to Bizer, Heath and Berners-Lee (2009), vocabularies are used to describe entities in the world, from different domains, and how entities relate to each other. Vocabularies are collections of classes and properties, expressed in RDF and following the guidelines of OWL and RDF. Anyone can create a vocabulary for Linked Data. The vocabularies can be related to other vocabularies by RDF triples wich link classes and properties (BIZER; HEATH; BERNERS-LEE, 2009).

There are several vocabularies, from different domains and with different objetives, such as, e.g., FOAF[2] for linking people and information on the Web, Good Relations[3] for details about product and service offerings, Dublin Core[4] provides meta-data to page annotations, Org[5] describes organizational structures in several domains, among others.

2.1.5 Linked Data

With the advent of the Semantic Web and the benefit of the data being understood by humans and machines, arises the Linked Data concept. This environment allows multiple software agents to process published content to provide information to people, using the Web as platform in order to create links between different data sets, maintained by different organizations in different locations (BIZER; HEATH; BERNERS-LEE, 2009). The data are interconected by URI's, published in RDF files and can be extracted by SPARQL Queries in the SPARQL endpoints related to data sets.

According to the W3C (2018a), the Semantic Web is a Web of Data. It provides not only access to data, but also to the relationship between them, hence the term Linked Data. Figure 3 shows the LOD Cloud in the year 2018, each circle representing a published data set whereas lines show their interconnection. Published data refers to several domains such as Media, Geography, Publications, Life Sciences, Cross Domain, among other. Each data set can have both incoming and outgoing links.

Some well-known example of data sets are: DBPedia[6] (cross domain), GeoNames[7] (geography), DBLP[8] (publications), PRotein Ontology[9] (life sciences), Linked Movie

2 <http://xmlns.com/foaf/spec/>.
3 <http://www.heppnetz.de/projects/goodrelations/>
4 <http://dublincore.org/documents/dcmi-terms/>.
5 <https://www.w3.org/TR/vocab-org/>.
6 <http://dbpedia.org>.
7 <http://www.geonames.org/ontology/documentation.html>.
8 <https://dblp.uni-trier.de/>.
9 <https://pir.georgetown.edu/pro/>.

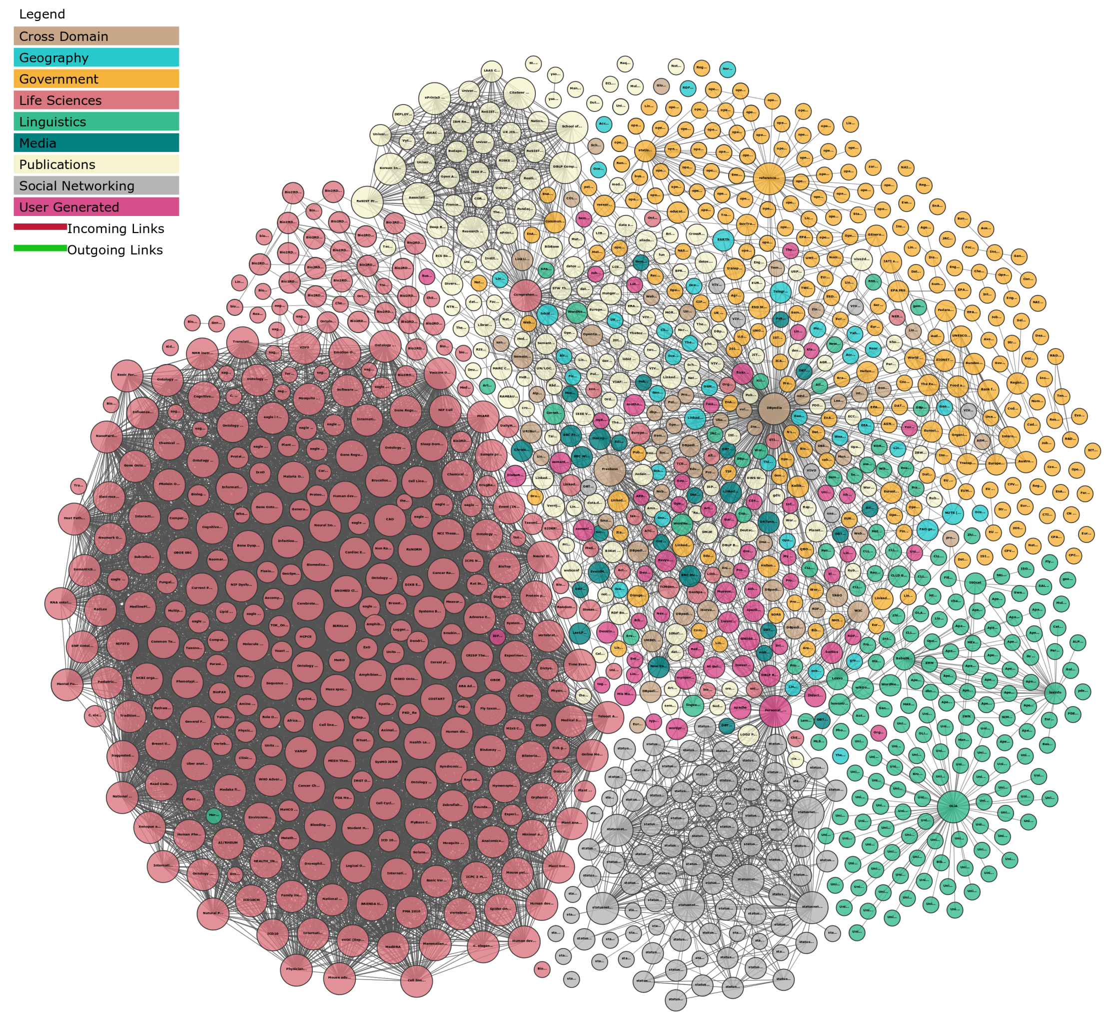

Figure 3 – The Linking Open Data Cloud Diagram available in <http://lod-cloud.net/>.

DataBase[10] (media), etc.

2.2 GORE (Goal-Oriented Requirements Engineering)

According to Lapouchnian (2005), GORE approaches seek to solve some deficiencies of the traditional Requirements Engineering (RE) approaches. There is an inadequacy of the traditional systems analysis approaches to deal with complex software; some techniques concentrate on the modeling and specification of the software alone, not being able to reason about the (composite) system and its environment. In general, GORE approaches focus on activities that precede the formulation of software requirements. The main activities are goal elicitation, goal refinement and various types of goal analysis, and the assignment of responsibility for goals to agents.

GORE-based methods focus on why systems are built, providing motivation and

[10] <http://www.linkedmdb.org/>.

justification for software requirements. The context analysis determines the reasons why systems should be constructed and their purposes according to different points of view (ANTON, 1996).

According to Lamsweerde (2001), goals are important in the RE process: they can be applied to different systems, cover functional and non-functional requirements and can be applied at different levels of abstraction. Systems (which cover software and its environment, composed of hardware, people and other systems) should aim to achieve their goals in their operations. Goals are important in RE processes because they assist in the requirements specification task, provide criteria for a complete analysis and to define which requirements are relevant; justify the importance of requirements to stakeholders; help to detect and eventually resolve conflicts between requirements, etc.

Lamsweerde (2001) explains that goals can be modeled according to intrinsic features such as *type*, for example, functional goals wich are services the system should offer and non-functional goals are expected qualities of the system, such as security, usability, safety, among others; *attributes* such as name and their specification, priority, utility and feasibility; and their *links* to other goals and other elements of the model, for example, for example, AND/OR refinement.

There are many different GORE approaches. The NFR Framework (MYLOPOULOS; CHUNG; NIXON, 1992) aimed at modeling and analysis of goals related to non-functional requirements. Tropos (BRESCIANI et al., 2004) uses iStar and seeks to support all analysis and design activities in the software development process, from domain analysis to implementation. KAOS (LAMSWEERDE; LETIER, 2000) has the objective of supporting the elaboration of requirements, provides a specification language, elaboration method, tool support and obstacles analysis which can cause obstruction of the goal. Techne (JURETA et al., 2010) seeks to support the representation and reasoning on instances of problems related to the requirements and propose solutions for each of these instances. GBRAM (ANTON, 1996) proposes to identify, elaborate, refine and organize goals for the requirements specification, also identifying goal obstacles; among other. In this work the particular focus is the iStar framework, addressed in the following subsection.

2.3 iStar (formerly i*)

iStar modeling (YU, 2009) seeks to understand social concepts and to apply them in systems engineering processes, in activities performed by analysts and designers when developing a system. iStar can be used in requirements engineering, enterprise engineering, security, privacy and trust modeling, among other areas.

The central concept is the *actor*, which can be human beings, organizations, hardware, software or a combination thereof. The actor is able to act independently, has

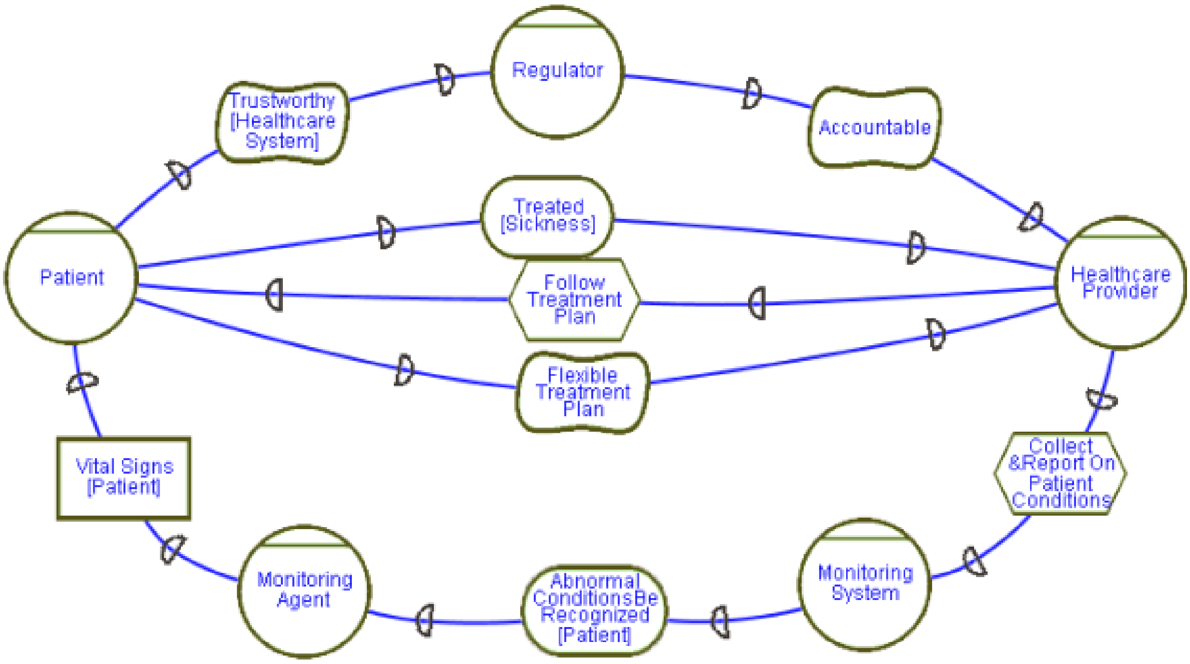

Figure 4 – A Strategic Dependency (SD) Model (YU, 2009).

autonomy, intention to perform an action and her behavior is not totally controllable and knowable. Actors have motivation and intention related to their behavior, to achieve their goals. Other concepts such as *tasks*, *resources*, *goal*, *softgoal*, *agent*, *roles*, etc. are part of this approach. iStar proposes two types of diagrams: *Strategic Dependency (SD)* and *Strategic Rationale (SR)* (YU, 2009).

According to Yu (2009), in the *Strategic Dependence Diagram (SD)*, the relationship of dependence is addressed: one actor (the *depender*) can depend on another (the *dependee*) for something (the *dependum*). The types of dependence are *goal dependency*, when there is a dependency on a goal to be achieved; *task dependency*, when there is dependency on a activity to be performed; *resource dependency*, when there is a dependency on a resource, which can be an information or material object, that will be consumed to reach the target; and *softgoal dependency* when there is a dependency on a quality, such as secure, privacy, reliability, among others.

Figure 4 illustrates these dependencies using an example from the health care domain (YU, 2009). For instance, the actor *Patient* depends on the *Healthcare Provider* to be *Treated [Sickness]* (a goal of the patient). On the other hand, the *Healthcare Provider* depends on the *Patient* to *Follow Treatment Plan* (a task representing exactly what the patient must do). Figure 5 shows the graphical representation of each iStar element.

According to Yu (2009), in the *Strategic Rationale Diagram (SR)*, it is possible to reason about the intentional elements that an actor wants to achieve, as well as to indicate how they can be achieved, this model focuses on external relationships. In this diagram goals, softgoals, tasks and resources are attributed to each actor. The *means-ends* link

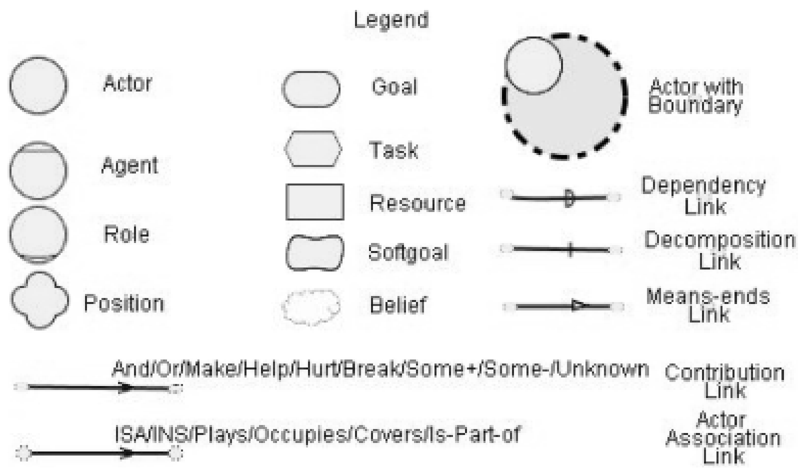

Figure 5 – iStar elements and their graphical representation (YU, 2009).

is used to connect a task to a goal (specifying ways they can be achieved), in turn tasks can be decomposed through *task decomposition links* to indicate the subtasks, subgoals, resources, and softgoals that need to be resolved (performed, satisfied, provided) for the task to succeed. Another possible relationship is the *contribution link* between a task and a softgoal, indicating how the former can contribute, positively or negatively, to the satisfaction of the latter.

Still in the health care domain, Figure 6 shows a Strategic Rationale Diagram, with three actors: *Patient*, *Helthcare Provider* and *Patient Assitant Software Agent*. These actors have goals and subgoals, perform tasks and use resources to achieve those goals. For instance, the actor *Patient* has the goal *Keep Well*, as well as some tasks to perform, such as *Restrict Access*, which contributes positively to softgoal *Confidential Medical Data*, the task *Patient Centred Care* is decomposed into *Plain Life Activities* and *Follow Customized Treatment Plan*, and this depends of resource *Customized Treatment Plan*.

Therefore, the iStar modeling language assists in the analysis of requirements, actors, goals, softgoals, dependencies, tasks and resources, supporting the development of systems. iStar allows you to check dependency between some elements and reason about the models. However, iStar was introduced in the 1990s (*i** at the time) as an actor-oriented modeling language and reasoning framework. Although it is widely used by the research community, it does present some difficulties, such as (DALPIAZ; FRANCH; HORKOFF, 2016):

- New users have difficulties to learn the language;
- Teachers do not have a shared knowledge to teach;
- Professionals do not receive an established reference to use iStar in their projects;

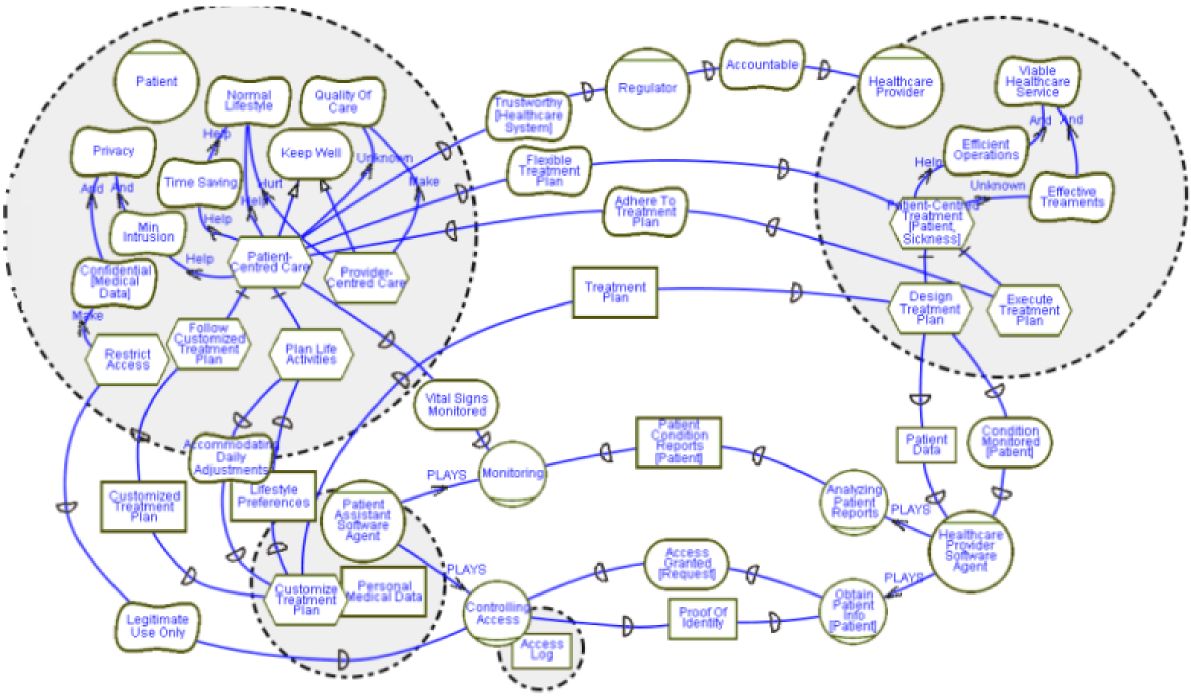

Figure 6 – A Strategic Rationale (SR) Model (YU, 2009).

- Technology providers cannot easily determine what are the main constructs to be implemented and the techniques to be applied on top of them.

These problems motivated the elaboration of a version 2.0 of the language, which was officially rebranded *iStar* (the * imposed some problems with respect to finding i* material on the Web). According to Dalpiaz, Franch and Horkoff (2016), in the iStar 2.0 language there are two types of actor: *Role* and *Agent*. The associations between them are *is-a* and *participates-in*. Is-a represents the concept of generalization / specialization, only roles can be specialized into roles, or general actors into more specific actors, e.g., *PhD Student is-a Student*. Agents cannot be specialized via *is-a*. The relation *participates-in* represents any kind of association different than generalization / specialization, between two actors. This association may exist between actors of the same type, from agent to agent, or different type, from agent to role, e.g., *agent Mike White participates-in role PhD Student*.

Still according to Dalpiaz, Franch and Horkoff (2016), the intentional elements are *Goal*, *Task*, *Resource* and *Quality*, the latter corresponding to the softgoal concept in previous versions. In iStar 2.0, the n-ary relationship called *Refinement* is used to link goals with tasks hierarchically. The types of refinements are *AND* and *OR*. Contribution links are used to link intentional elements on qualities, their types being *Make*, *Help*, *Hurt* and *Break*. The *NeededBy* relationship is used to link a task with a resource and indicates that the resource is required to perform the task. The *Qualification* relationship occurs between qualities with a task, goal or resource, for example the quality *No Errors*

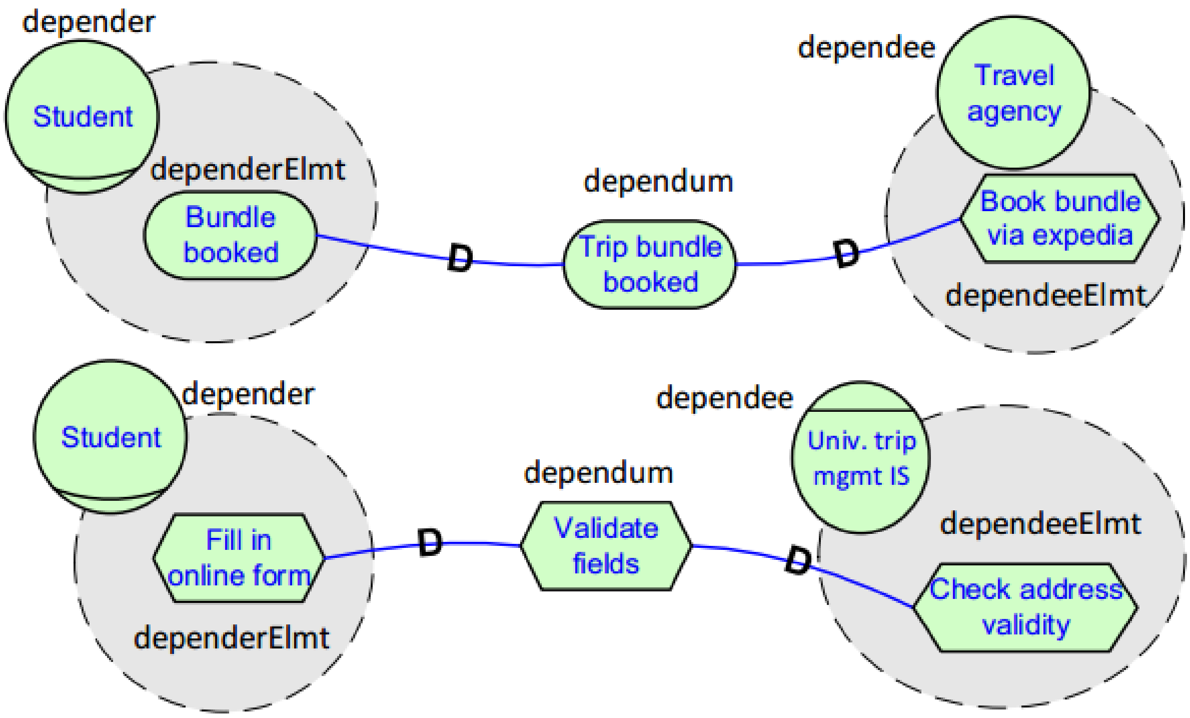

Figure 7 – Examples of iStar 2.0 dependencies (DALPIAZ; FRANCH; HORKOFF, 2016).

qualifies the goal *Request prepared*, because for a request to be closed there should be no errors. Regarding social dependencies, iStar 2.0 has five elements: *depender*, *dependerElmt*, *dependum*, *dependee* and *dependeeElmt*, illustrated in Figure 7. The types of dependum are *goal*, *task*, *resource* and *quality*.

2.4 Risks Concepts

According to Bannerman (2008), the most common definition of risk in software projects is the exposure of specific factors that pose a threat to the achievement of expected outcomes in a project. There are two viewpoints about the risks, between academics and practitioners. Traditionally, risk is defined as chance or probability with negative consequences, such as injury or loss. In the other view, the risk can positively or negatively impact on the project goals (LEHTIRANTA, 2014).

In (BOEHM, 1991), the formula $RE = Prob(UO) * Loss(UO)$ is used to define the risk exposure, where $Prob(UO)$ is the probability of an unsatisfactory outcome and $Loss(UO)$ is the loss to the parties affected if the outcome is unsatisfactory. The risk factors must be identified at the beginning of the project. Control measures such as mitigation strategies and/or contingency plans can be used to minimize the likelihood of occurrence of a risk event or the impact in case of occurrence (BANNERMAN, 2008).

Risk management in software projects is important because it aims to identify, analyze and manage risk factors in order to increase the chances for the outcome of the project to be successful and/or avoid failures in the project (BANNERMAN, 2008). The steps of risk management are *Risk Identification*, *Risk Analisys*, *Risk Prioritization*, *Risk Management Planning*, *Risk Resolution* and *Risk Monitoring* (BOEHM, 1991).

For this work, the steps *Risk Identification* and *Risk Analysis* are very important. Risk identification produces a list of project-specific risks that can compromise a project's satisfactory outcome. Some identification techniques are checklists, decomposition, comparison with experience, and examination of decision drivers (BOEHM, 1991). According to (WESTFALL; ROAD, 2001), there are many techniques for identifying risks, including interviewing, reporting, decomposition, assumption analysis, critical path analysis and utilization of risk taxonomies.

Risk Analysis seeks to assess the probability of loss and loss magnitude related to each of the identified risks. Some techniques include network analysis, decision trees, cost models, performance models, and statistical decision analysis (BOEHM, 1991). In the analysis phase, risk data is converted into information for the decision making, it includes reviewing, prioritizing, and selecting the most critical risks to address. Each risk can be assessed according to cost, schedule, performance and product quality (AL; CHOWDHURY; AREFEEN, 2011).

According to Bannerman (2008), the objectives of risk response strategies are to reduce or eliminate the likelihood of the threat occurring, limit the impact of the risk if it is realized or a combination of both. The four most common are: *Avoidance*, *Transference*, *Mitigation* and *Acceptance*.

Avoidance seeks to avoid the occurrence of a negative effect or impact on the project; *Transference* transfers responsibility for risk to third parties, but does not eliminate the risk; *Mitigation* corresponds to one or more reinforcement actions to reduce the threat to the project, reducing its likelihood and/or impact before the risk is realized; and *Acceptance* means accepting that the risk exists, doing nothing, monitoring the status of the risk, and if the risk is realized execute a contingency plan (BANNERMAN, 2008).

2.5 RISCOSS Approach

The *RISCOSS* project (COSTAL et al., 2015) came about because of the growing adoption of OSS (*Open Source Software*) by organizations. The occurrence of risks in the adoption of OSSs can impact the business goals of the organization.

In RISCOSS, the modeling of risks is done using *RiskML* (COSTAL et al., 2015), a language that uses primitive concepts like *Goal* — something of interest for a stakeholder to obtain or maintain; *Event* — the occurrence of something that may undermine the objectives; *Situation* — circumstances where risks are likely to occur; and *Indicators* of

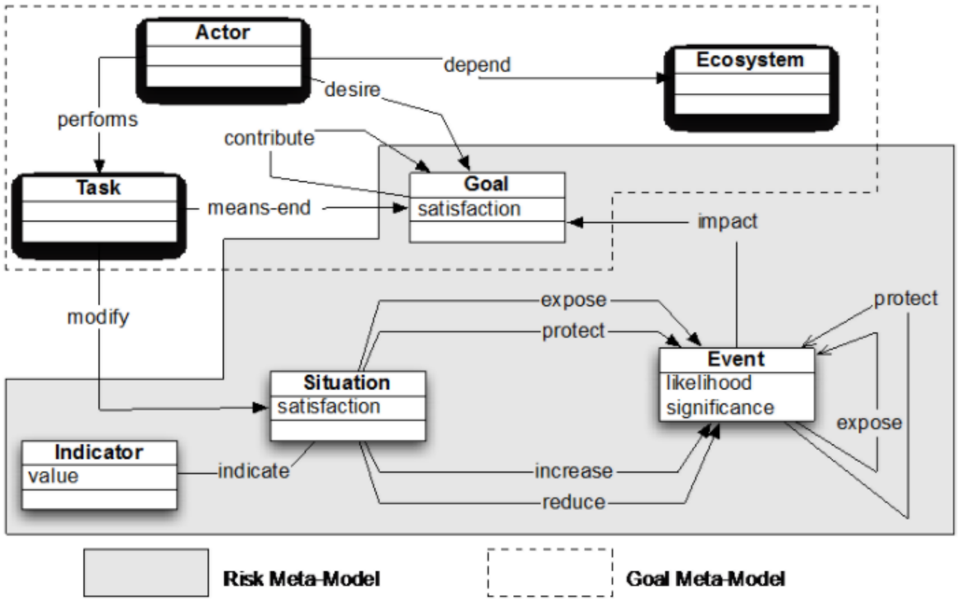

Figure 8 – RiskML-iStar Integrated Metamodel (LÓPEZ; SIENA, 2015).

risks — existing data measurements approved by experts (LÓPEZ; SIENA, 2015), which can be simple or composite. Moreover, the impact relationship, between an event E and a goal G, indicates that the occurrence of event E impacts on the satisfaction of G. The attributes of RiskML's primitive constructs are *Likelihood* (Event), *Significance* (Event), *Exposure* (Event), *Satisfaction* (Situation) and *Satisfiability* (Goal); and relationships are *indicate* (between Indicator and Situation); *expose*, *protect*, *increase*, *reduce* (between Situation and Event); *expose* and *protect* (between Events); and *impact* (between Event and Goal) (LÓPEZ; SIENA, 2015).

RISCOSS is also founded on Goal-Oriented Requirements Engineering (GORE), using the iStar language (c.f. Section 2.3) to build goal models and integrating GORE concepts (such as *Actor*, *Task* and *Ecosystem*), including *Goal*, which is also a concept for RiskML and provides a way to integrate risk and goal models. Figure 8 shows the meta-model for such integration (LÓPEZ; SIENA, 2015).

In RISCOSS (LÓPEZ, 2015), risk management is based on a three-layered strategy, depicted in Figure 9. The layers cover the gathering of data. In layer 1, data about risks is collected from OSS communities, projects and experts that determine the risks drivers; in layer 2, risk indicators are defined and models in RiskML language are created; and in layer 3, the risk model is linked with the the goal models to represent the impact that the possible risk events have on strategic and business goals.

2.5.1 Goal and Risk (separate) Models

The first step, in the RISCOSS approach, is to create goal models. For instance, Figure 10 shows the business goal model for the organization TEI, which is a part of

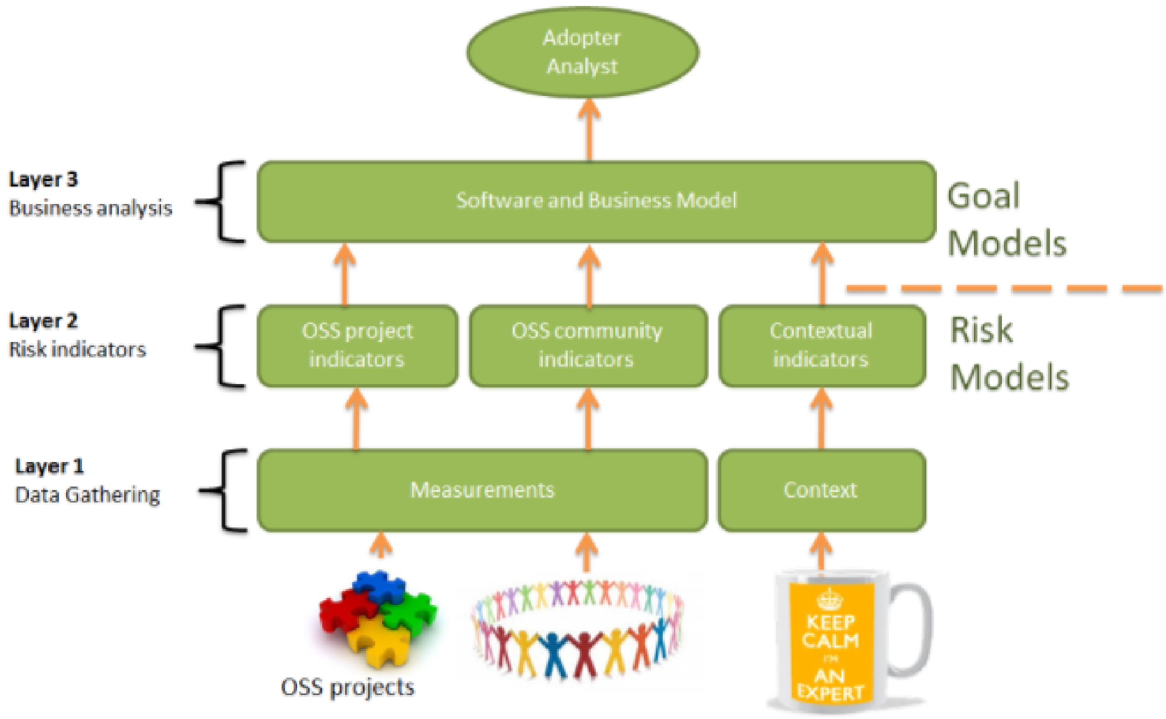

Figure 9 – RISCOSS Three-layered Strategy (LÓPEZ, 2015).

Ericsson, related to product maintenance, with some OSS components. The actor *Ericsson* has the main goal *Time-to market reduced* which depend on the actor *TEI*, which has business goals such as *Costs reduced*, *Development time reduced* and softgoals *Product requirements achieved*, *Maintainable code*, *Quality of code*, *Mature technology used* and *Secure code*.

TEI, in turn, depends on the *OSS Community* to provide resources *OSS component code* and *OSS component documentation*; and to satisfy the goal *Support obtained*. Also, *TEI* depends on the actor *3PP OSS Provider* to satisfy the goal *OSS component maintained*, and this depends on the *OSS Community* to deliver resource *OSS component code + docum*(entation).

Next, risk models are created. Figure 11 shows a risk model for OSS code maintainability using the RiskML language. In this model, new goals can be added (with respect to the goal models produced beforehand), as well as the situations that expose the risks events, which in turn impact on the goals related to code maintainability. According to Costal et al. (2015), the current risk model is based on interviews with managers, literature study on OSS risks and various metrics for code quality, such as complexity metrics, lines of code, and test coverage.

For example, the situation *Unavailability of commun. or ext. maintenance* exposes the risk event *No proficient maintainer available*, which impact the goals *Rely on 3rd party maintenance* and *Obtain maintenance from the community*.

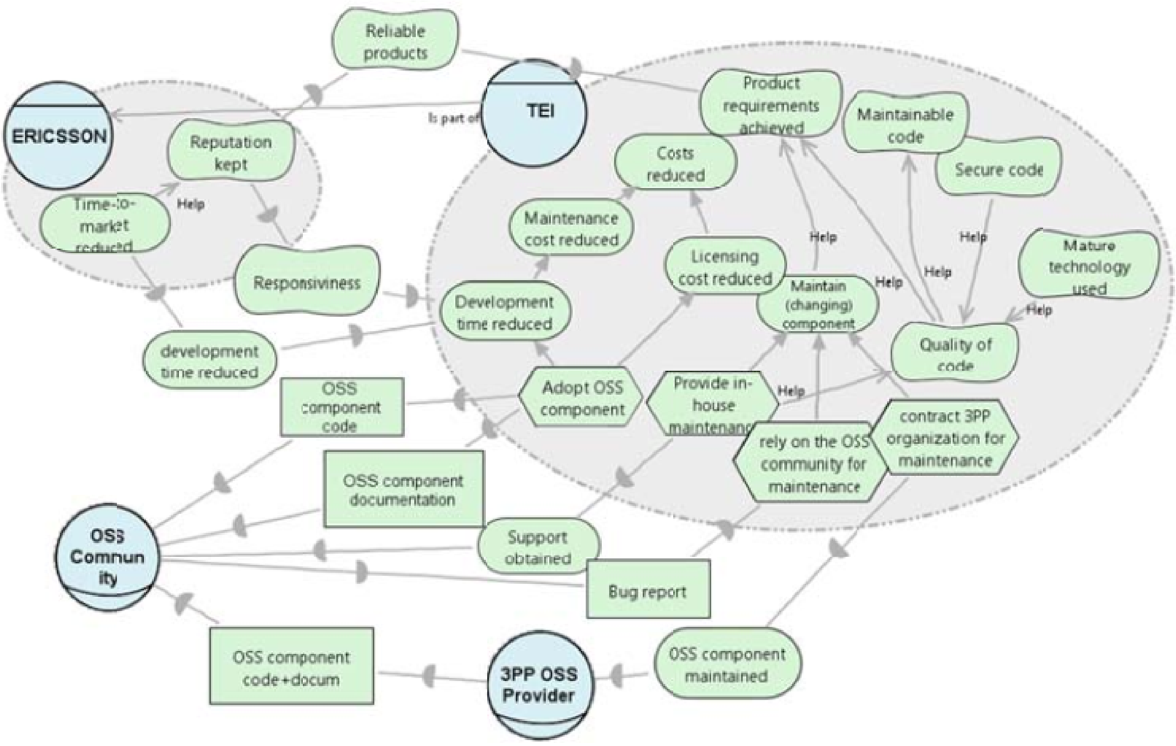

Figure 10 – TEI - Business goal model (COSTAL et al., 2015).

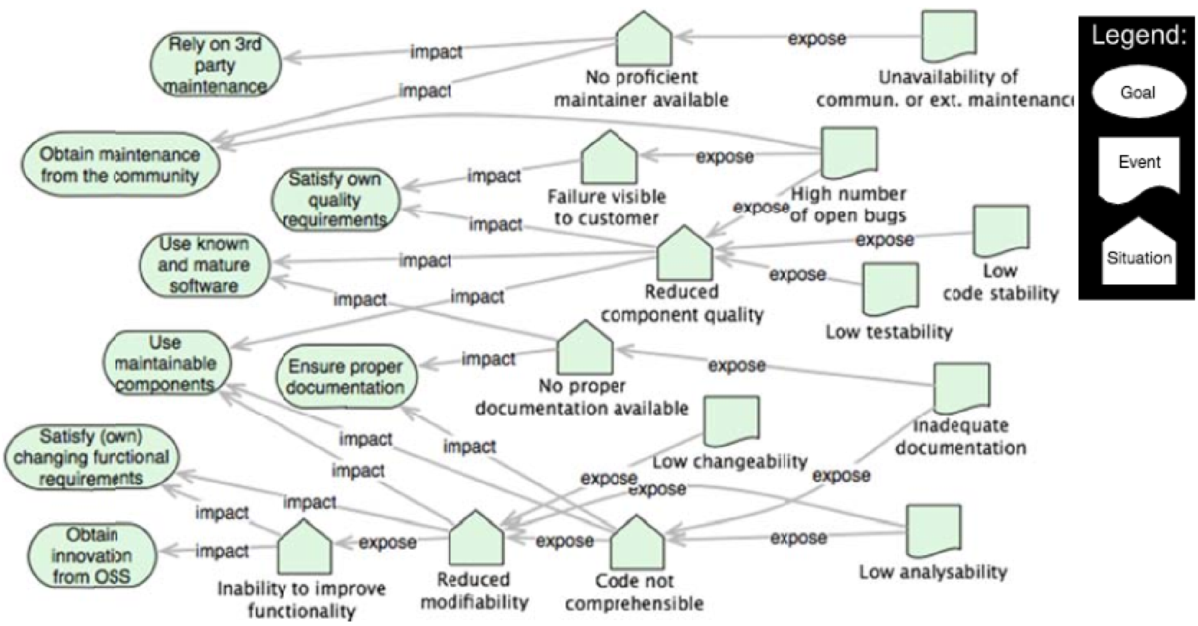

Figure 11 – RiskML - Risk Model for OSS Code Maintainability (COSTAL et al., 2015).

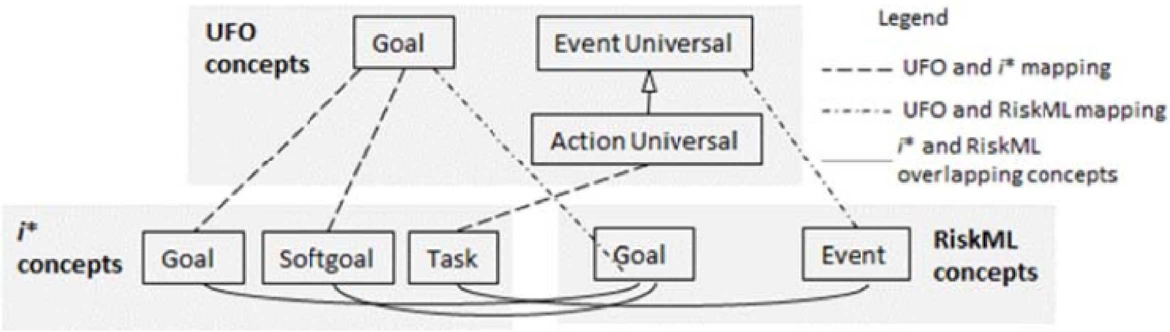

Figure 12 – iStar and RiskML - Overlaping Concepts According to UFO Mappings (COSTAL et al., 2015).

2.5.2 Goal and Risk Integrated Model

Afterwards, the goal model is integrated with the risk model. According to Costal et al. (2015), for this integration it is necessary to identify concepts that have the same semantics between iStar and RiskML. This analysis must be trimmed by ontologies and in this case the Unified Foundational Ontology (UFO) (GUIZZARDI, 2005) is used.

In Figure 12, *Goal* and *Softgoal* in iStar and *Goal* in RiskML are mapped to the UFO concept of *Goal*. An *Event* in RiskML overlaps *Task* in iStar, meaning these two concepts map under certain conditions: (1) the iStar *Task* is mapped to UFO *Action Universal* and RiskML *Event* is mapped to UFO *Event Universal*; and (2) in UFO, only *Events* deliberately performed by agents to fulfill their intentions are considered *Actions*. Therefore, iStar *Tasks* are mapped to *Events* in RiskML but, not every *Event* in RiskML can be mapped to an iStar *Task* (COSTAL et al., 2015).

Based on the above ontological analysis and given the *impact* relationship, in which an occurrence of an event E impacts on the satisfaction of a goal G, Costal et al. (2015) propose a RiskML–iStar Integrated Metamodel, shown in Figure 13 (which is a more detailed version of Figure 8). In this metamodel, *Goal*, *Task*, *Softgoal* and *Resource* are subtype of *IntetionalElement*, and impacted by a *Event* from RiskML. The author proposes that *Tasks* and *Resources* are *IntentionalElements* affected by an *Event*, because *Tasks* are interpreted as UFO *Action Universals*, intentional actions with the aim of satisfying a *Goal*. In the case of *Resources*, they are used to reach a *Goal*, therefore we can assume the existence of a related goal.

According to Costal et al. (2015), given a business (goal) model B which contains a set of intentional elements I, and given a risk model R containing a set of goals G, what we want is to integrate the two models to produce a model M, in which the goals in G are semantically connected to the intentional elements in I, according to the ontological alignment presented earlier. The proposed alignment cases are:

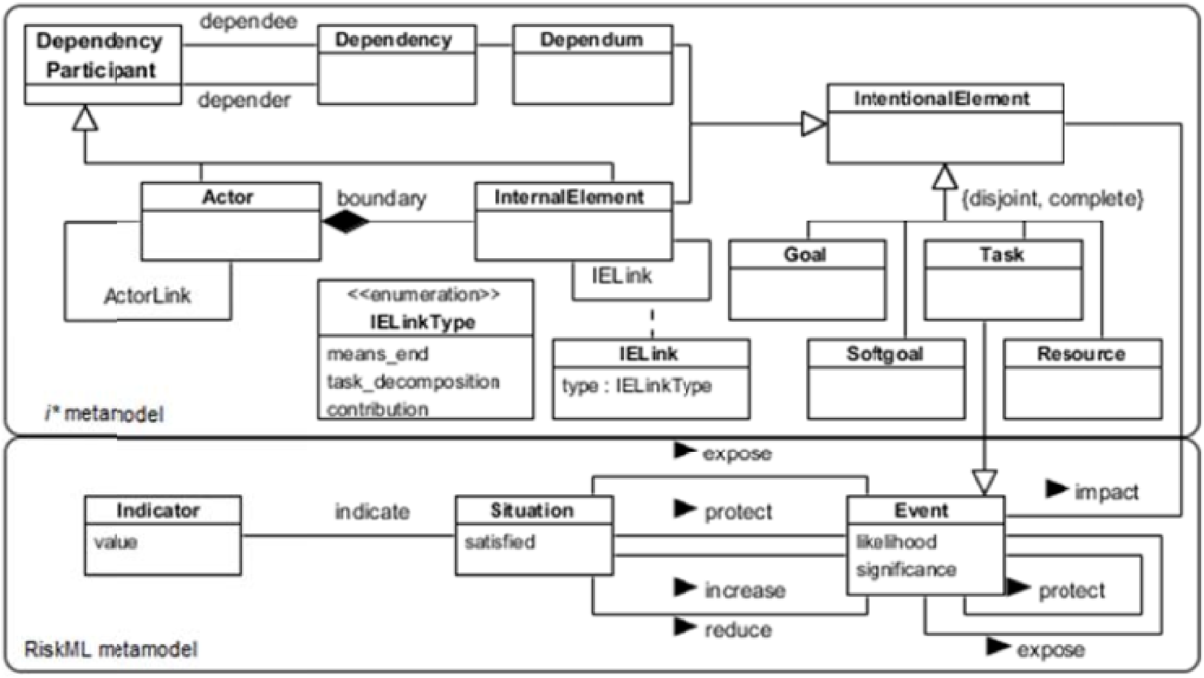

Figure 13 – RiskML-iStar Integrated Metamodel (COSTAL et al., 2015).

- **Alignment case 1**: there exists an intentional element X in B, such that G can be semantically equivalent to it. In this case, the resulting model M will keep X and will include the impacts from events in R to G, but replacing X by G;
- **Alignment case 2**: there exists an intentional element X which subsumes G, in which case the model M will include both, relating them appropriately. For example, using means-end if X is a goal, or contribution link if it is a softgoal;
- **Alignment case 3**: in case there is no G in I that satisfies alignment cases 1 or 2, this means that there is no obvious impact from the risk to any business goal. In this case, it is necessary to interact with the business analysts with two possible outcomes: (i) G refers to a business goal that was not raised in the initial version, in this case the business model must be updated to fall into cases 1 or 2; or (ii) G represents a risk that is not important for the company, in which case G, as well any element in R which refers only to G, is not included in M.

Figure 14 shows an example of integrated model, result of applying the alignment cases above using the goal model of Figure 10 and the risk model of Figure 11. As explained by Costal et al. (2015), Table 1 details the alignment of these two models. For example, goals from *g1* to *g4* are equivalents (Alignment case 1) and the elements from are B are kept. Goals *g5* and *g6* can be considered subsumed by one element of B, and the elements of R are included in B as goals. Finally, goal *g7* is included in the TEI business model, while *g8* is discarded.

Table 1 – Example of alignment of Business and Risk models (COSTAL et al., 2015).

	g in R	x in B	New link
	Alignment case 1 (equivalent)		
g1	Obtain maintenance from the community	Rely on the OSS community for maintenance	
g2	Rely on 3rd party maintenance	Contract 3PP organization for maintenance	
g3	Use known and mature software	Mature technology used	
g4	Use maintainable components	Maintainable code	
	Alignment case 2 (subsumes)		
g5	Satisfy (own) changing functional requirements	Product requirements achieved	contribution link
g6	Satisfy own quality requirements	Product requirements achieved	contribution link
	Alignment case 3.a (new business goals)		
g7	Ensure proper documentation	Ensure proper documentation (new)	task-decomposition (Adopt OSS component), contribution link (Maintainable code)
	Alignment rule 3.b (discarded)		
g8	Obtain innovation from OSS		

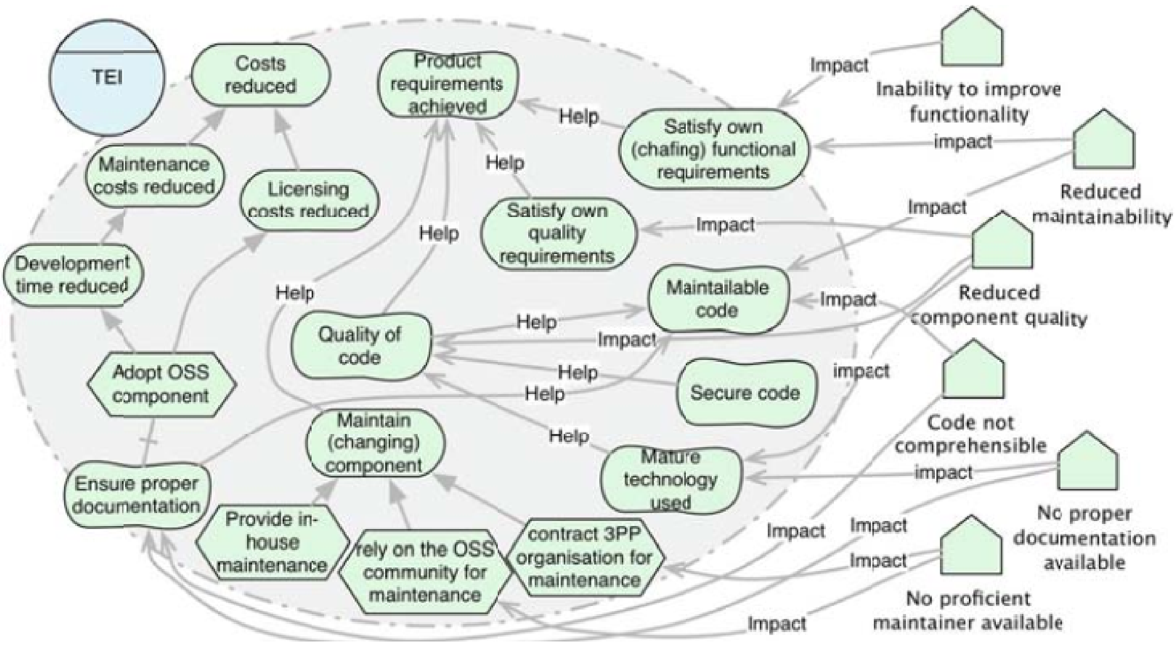

Figure 14 – TEI SR Diagram Connected to RiskML Risks Events (COSTAL et al., 2015).

The RiskML-iStar integrated model can finally be used to support risk analysis. The impact relation between a risk and a goal represents a negative effect when the event is likely and significant, increasing the evidence that the goal is not achieved. Such evidence can then be propagated through the goal graph calculating, for each intentional element, if it is totally/partially satisfied/denied, enabling us to see how risks affect the strategic/high-level goals of each of the involved actors and prioritize our risk mitigation efforts based on this analysis (COSTAL et al., 2015).

2.6 FrameWeb-LD Approach

FrameWeb-LD (CELINO et al., 2016) is an approach for building Web-based Information Systems (WIS) that publish Linked Data. It proposes a process divided in five stages: *Analysis*, *Design*, *Implementation*, *Testing* and *Deployment*. The main contributions of this approach are an extension of *FrameWeb*'s metamodel (MARTINS; SOUZA, 2015) allowing Linked Data mappings to be represented in its design models, and a tool for code generation to assist developers in publishing Linked Data. Figure 15 shows a overview of the FrameWeb-LD Proposal, the phase names are on the right side of the figure.

In the following subsections, the phases and activities of the FrameWeb-LD approach will be addressed, with particular focus on the activities used in the GRALD proposal, presented in Chapter 3.

2.6.1 Analysis Phase

In the analysis stage, there are two activities: *Elicit Requirements* and *Develop Domain Model in OntoUML*. FrameWeb-LD suggests SABiO (FALBO, 2014) for eliciting requirements and developing a domain model in OntoUML (GUERSON et al., 2015). SABiO is a Systematic Approach for Building Ontologies, which comprises five stages: (1) purpose identification and requirements elicitation; (2) ontology capture and formalization; (3) design; (4) implementation; and (5) test.

During purpose identification and requirements elicitation, it is necessary to identify the ontology purpose and its intended uses, then to perform the elicitation of requirements. These requirements can be divided into functional and non-functional requirements. Functional requirements refer to the content to be represented by the ontology and are usually written as CQ's (Competency Questions), i.e., questions that the ontology is supposed to answer.

Ontology capture and formalization can be performed with the aid of tools such as OLED[11] or Menthor.[12] According to Falbo (2014), the objective of this activity is

[11] <http://nemo.inf.ufes.br/projects/oled/>
[12] <http://www.menthor.net/>

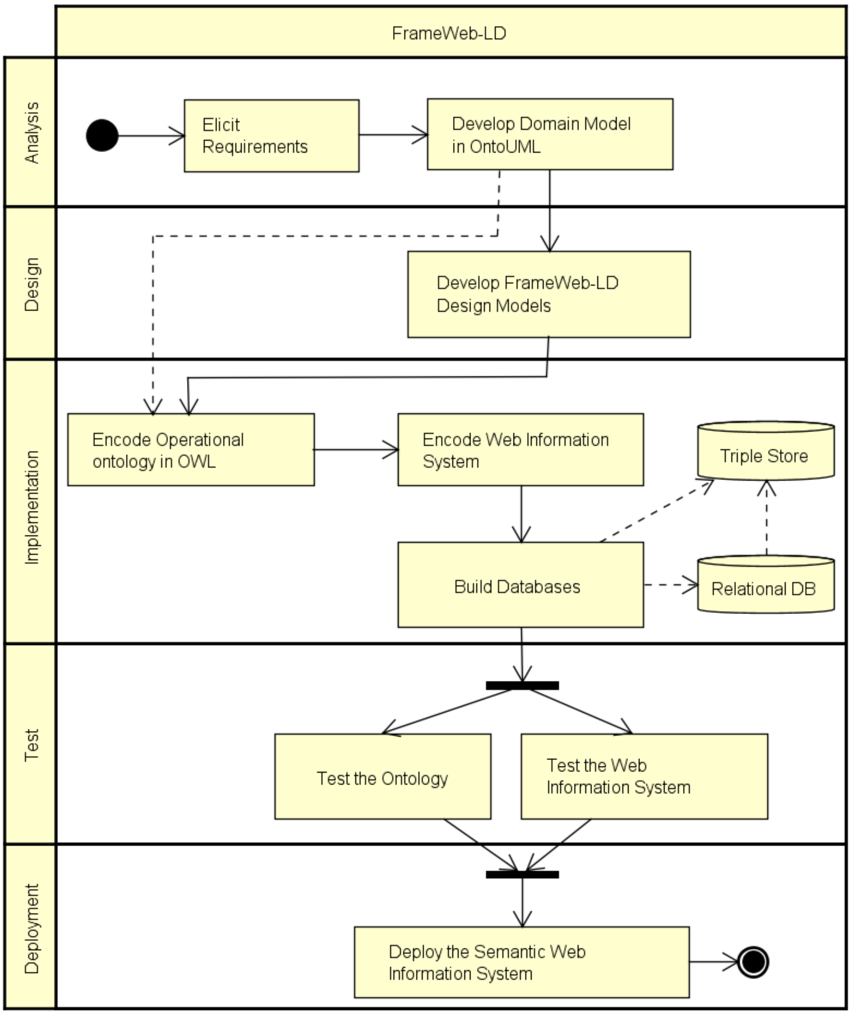

Figure 15 – Overview of FrameWeb-LD Proposal (CELINO et al., 2016).

to establish a shared conceptualization of the domain, seeking to establish a consensus between domain specialists, where relevant concepts and relations must be identified and organized, guided by the competency questions elicited in the previous phase. The ontology model is based on UFO (GUIZZARDI, 2005) and the modeling language must guarantee the expressiveness of the ontological models, strongly axiomatized. The class diagrams are created with OntoUML, which is based on UML but incorporates UFO-based distinctions.

2.6.2 Design Phase

In the design phase, *FrameWeb-LD's Entity Model* is created based on the conceptual model built in the previous phase. *FrameWeb-LD* adds annotations on top of the basic *FrameWeb Entity Model* (MARTINS; SOUZA, 2015) to specify linked data vocabulary mappings, based on FrameWeb-LD metamodel (CELINO et al., 2016). An example is shown in Figure 16, which represents the *Entity Model* for an academic Web-based Information System called Marvin,[13] under development in our department at the university.

[13] https://github.com/dwws-ufes/marvin/

In particular, we focus on a module of Marvin called C2D, which keeps track of members of our postgraduate program and their respective publications for evaluation purposes.

Figure 16 shows UML classes representing researchers, publications, etc. linked to popular vocabularies, such as FOAF and DBLP.[14] For instance, the `Researcher` class is equivalent to class `dblp:Person`, given that the scope of the DBLP vocabulary is to represent researchers and their publications. According to Celino et al. (2016) the *subclass* relation between the vocabulary class and the domain class can be represented by inheritance, e.g., the class `Researcher` is subclass of `foaf:Person` (FOAF has a broader scope and represents not only researchers). The *subPropertyOf* denotes relations between properties, in this example, the association between `Publication` and `Venue` is *rdfs:subPropertyOf dblp:publicationType*. Finally, in the `User` class, the `ld-ignore` stereotype represents that user data will not be published as Linked Data.

2.6.3 Implementation, Test and Deployment Phases

The *Implementation* phase contains three activities: *Encode Operational Ontology in OWL*, *Encode Web Information System* and *Build Databases*. The first activity can be performed with the aid of code generation functionality present in editors such as Menthor or OLED. Starting from this generated OWL file, a prototype tool named *ReMaT* (Relational Database Mapping to Triple Store) generates the mapping from the classes that represent concepts of the system with the external vocabularies represented in the *Entity Model* (CELINO et al., 2016).

The codification of the Web Information System follows the usual process for FrameWeb, whereas for the last activity, a relational database is created (also following usual software development routines) and a Linked Data layer above it is added with the use of an RDF-over-Relational Database layer such as D2RQ,[15] which provides triplestore (a database of RDF triples) features such as a SPARQL endpoint.

After the Implementation phase is performed, the phases *Test* and *Deployment* are carried on, with the deployment of Linked Data done by D2RQ.

2.7 Conclusion of this Chapter

In this chapter, a review of the literature was presented, containing concepts about Semantic Web and Linked Data; risks in software engineering; GORE approaches, with more emphasis in iStar; as well as the RISCOSS and FrameWeb-LD approaches.

RISCOSS, for Open Source Software, uses goal modeling with iStar, and risk

[14] <http://xmlns.com/foaf/spec/>, <http://dblp.uni-trier.de/>
[15] <http://d2rq.org/>

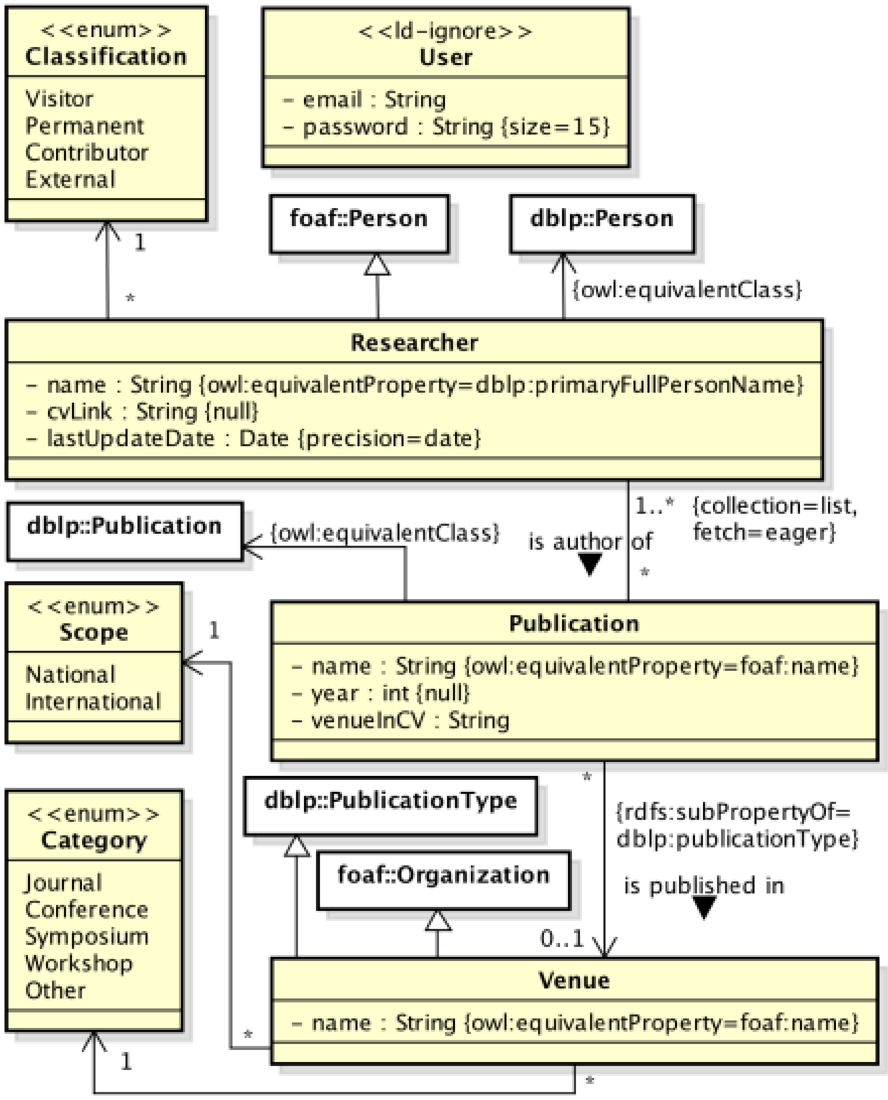

Figure 16 – A FrameWeb-LD Entity Model for C2D (CELINO et al., 2016).

modeling with RiskML, allowing to create models separately, then producing an integrated model, with the objective of analyzing the impact of a risk event on the goals of stakeholders. FrameWeb-LD focuses on Linked Data publication, including important activities for this work, such as *Elicit Requirements*, *Develop Domain Model in OntoUML*, *Develop FrameWeb-LD Design Models*, among others.

Based on this literature review, more specifically on the RISCOSS and FrameWeb-LD approaches, this work proposes *GRALD* (*Goal and Risk Analysis for Linked Data*), addressed in the next chapter.

3 Proposal

In this chapter, we present our proposal, named *Goal and Risk Analysis for Linked Data* (*GRALD*), presented for the first time in (FREITAS et al., 2018). The main objective of this work is to support developers of information systems on the Web in the analysis of goals and risks in the publication of Linked Data by these systems and in the choice of appropriate vocabularies. The main contributions of the approach are:

1. Modeling of system requirements using a goal-oriented language, with a particular focus on the publication of Linked Data;
2. Creation of risks models to support the analysis of risks in the publication of Linked Data;
3. Searching of vocabularies according to the elicited goals and conceptual models of the domain;
4. Providing a unified catalog of goals, risks, tasks and resources for Linked Data.

GRALD (*Goal and Risk Analysis for Linked Data*) is based on two existing approaches: RISCOSS and FrameWeb-LD. We chose RISCOSS because it uses two different languages for modeling goals (iStar) and risks (RiskML), which allows one to study how the same risks may affect different strategies or ecosystems (COSTAL et al., 2015). RISCOSS extends the goal analysis support in iStar, allowing us to analyze how risks are propagated in the goal graph. Frameweb-LD was chosen because it is focused on the publication of Linked Data in Web-based Information Systems. Through GRALD, we seek synergy between these two approaches.

With some effort, other approaches related to risks and goals could be adapted to use in GRALD, e.g., the GR Framework (ASNAR; GIORGINI; MYLOPOULOS, 2011) for modeling and reasoning about risks during requirements analysis or the KAOS language (LAMSWEERDE; LETIER, 2000) for goal modeling and obstacle analysis. This is, however, out of the scope of our work.

An overview of the development process proposed by *GRALD* is presented in Figure 17. This figure was created with Bizagi tool[1] using BPMN[2] notation. The process is divided in three stages (the names of the roles defined in each swimlane). Blue rectangles (light background) represent activities proposed in FrameWeb-LD (CELINO et al., 2016),

[1] <https://www.bizagi.com>
[2] <http://www.bpmn.org/>

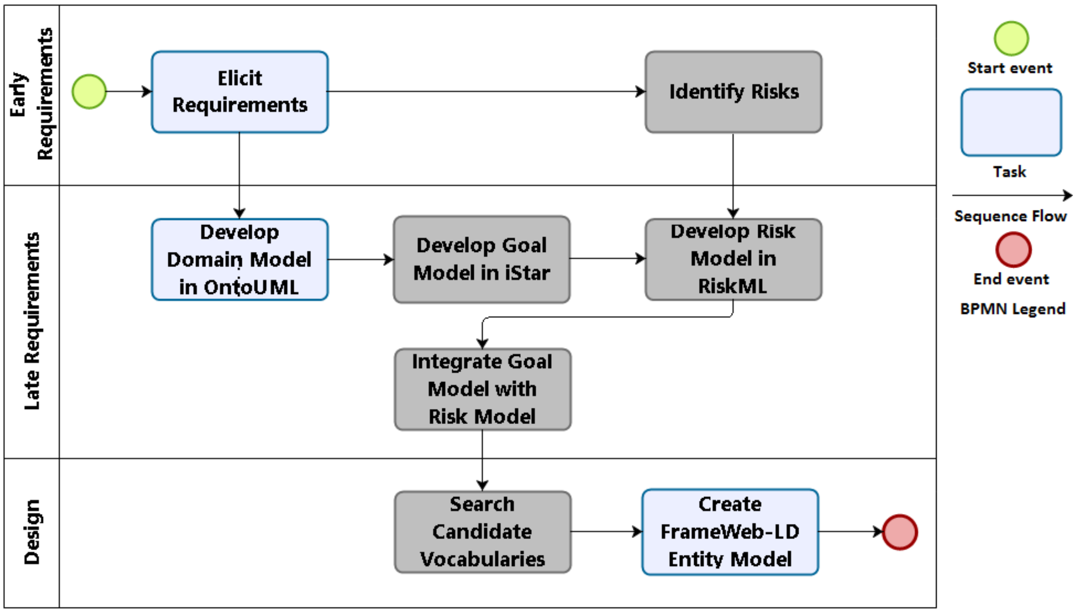

Figure 17 – Overview of the GRALD process.

presented in Section 2.6, whereas gray rectangles (dark background) represent activities proposed by *GRALD* (adapted from RISCOSS) to meet the above contributions. Arrows represent the sequences of activities. It is important to note that while the figure indicates a sequence of activities, we do not prescribe a specific development life-cycle. The phases of the process are detailed in the following subsections.

It is important to note that the catalog presented in the Section 3.4 can be used in the activities *Elicit Requirements*, *Identify Risks*, *Develop Domain Model in OntoUML*, *Develop Goal Model in iStar*, *Develop Risk Model in RiskML*, *Integrate Goal Model with Risk Model* and *Search for Candidate Vocabularies*. Given that it is used in so many activities, it was not represented in the figure to avoid polluting it with so many connections.

In our proposal we apply these approaches in a unified way, performing goal and risk modeling with iStar and RiskML (RISCOSS), respectively, for the publication of Linked Data with FrameWeb-LD. Thus, we seek synergy between these approaches to aid in the choice of Linked Data vocabulary to be used by a Web-based Information System, understanding the risks involved in the publication and integration of Linked Data.

3.1 Early Requirements

In this phase the activities performed are *Elicit Requirements* and *Identify Risks*, with the purpose of providing elements for the construction of models in the next phase, related to Domain Ontology, Goal and Risk models.

3.1.1 Elicit Requirements

The first phase starts with the *Elicit Requirements* activity, whose purpose is twofold: elicit requirements for the Web-based Information System (WIS) to be developed (i.e., its functional and non-functional requirements); and elicit requirements for an ontology of the domain in question (i.e., the data manipulated by the WIS). Also, these requirements are used to build iStar and related models in the next phase.

Requirements for the WIS should be elicited using any Requirements Engineering technique for early requirements and the iStar language could also be used for this purpose. As for ontology requirements, Celino et al. (2016) suggests the use of techniques prescribed by SABiO (FALBO, 2014) in order to identify the purpose and elicit the requirements for an ontology of the domain in question. Such requirements are then documented in the form of Competency Questions (CQs), which are questions the ontology should be able to answer.

In (CELINO et al., 2016), for the C2D system example introduced in Section 2.6, some of the elicited CQs were “What is a researcher in the post-graduate program?” (CQ1), “What are the possible roles for a researcher?” (CQ2), or “What is the scoring system to evaluate researchers in the program?” (CQ3), The answers obtained by these CQs serve as a basis for the creation of the Conceptual Model in OntoUML.

The input of this stage is knowledge about the domain of the system. The outputs of this stage are the competency questions and the requirements elicitation document.

3.1.2 Identify Risks

In this activity, Risk Identification is performed, as discussed in Section 2.4. Bibliographic references related to Linked Data are used to support this phase. According to Hyland, Atemezing and Villazón-Terrazas (2014), best practices for publishing Linked Data should be considered, such as choice of dataset; URI creation; choice and creation of vocabulary; choice of an appropriate license for the publication of content; among others. The W3C (2017) also addresses best practices related to Data on the Web. The adoption of these best practices helps prevent risks and, conversely, starting from them, we can identify possible risks related to the publication and consumption of Linked Data in our projects.

In (BRUWER; RUDMAN, 2015), traditional Web risks are extended to the Semantic Web, and specific risks of Linked Data and Semantic Web such as *SPARQL* and *SPARUL* injections, among others, are also analyzed. Risks related to the creation and maintenance of ontologies and trust and proof of information are also addressed (although not elicited for our running example).

Three questions were elaborated to conduct this activity, with the objective of identifying situations, risk events, and also new goals related to the Web and Linked Data systems. The questions are:

1. What are the risks of traditional Web applications that can impact the goals related to Linked Data and others goals of the WIS being developed?
2. What are the Linked Data risks related to vocabulary adoption, creation and maintenance of ontologies, publication of data in RDF format, trust and proof of information that can impact the goals related to publication of Linked Data in the WIS being developed?
3. What new goals can be added to the risk models, during the identification of risks, in the context of the WIS being developed?

The Risks are identified and separated by categories, according to the tasks performed and goals to be achieved, for example, vocabulary adoption, data publication, data provenance, among others.

In the RISCOSS project (LÓPEZ, 2015), risk management is based on a three-layered strategy, summarized in Section 2.5 (cf. Figure 9). We adapt this strategy to the case of Linked Data publication, collecting data about risks from the bibliography and Linked Data community websites. For example, in the context of C2D, Table 2 shows situations and risk events, as well as new goals related to vocabulary adoption. The adoption and choice of vocabulary is an important activity in Linked Data publication, because it defines how the published content will be represented. These risk events should be taken into account when choosing vocabularies. Further, Table 3 refers to situations, risks events and goals related to data publication. In the case of C2D, the main goal, related to Linked Data, is data publication.

As these situations come from aforementioned best practices documents, they may not vary a lot from a system to another. They will most likely depend on stakeholder preference and organizational context. Such similarity among projects motivated us to propose the catalog of risks and goals, presented later in Section 3.4.

The output of this stage are tge identified risks, as exemplified in tables 2 and 3 for the C2D system. Based on this risk identification activity, the risk models in the RiskML language will be created (cf. Section 3.2.3), focusing on the impact of risk events on the new identified goals, in the categories related to Linked Data systems, e.g., Vocabulary Adoption, Data Publication and Use of Tool or Software.

Table 2 – C2D RiskML Concepts — Vocabulary Adoption

RiskML - Goal	**RiskML - Event**	**RiskML - Situation**
Use documented and self-descriptive vocabularies	No proper documentation available or sufficient	Inadequate or nonexistent documentation
Use known vocabularies	Reduced popularity	Low vocabulary referencing
Use active vocabularies	Non-existent representation	Deprecated vocabulary
Obtain maintenance from publisher	No proficient publisher available	Unavailability of publisher
Use multilingual vocabulary	Multilingual vocabulary not provided	Vocabulary with only one language
Use open license for the vocabulary	Open license not provided	Vocabulary with proprietary license

Table 3 – C2D RiskML Concepts — Data Publication

RiskML - Goal	**RiskML - Event**	**RiskML - Situation**
Use Cool URI	URI in accordance to the best practices is not provided	URI in non-compliance with best practices
Access to RDF always available	Inaccessible site	Infrastructure problem
Data updated and accurate	Data not updated or incorrect	Wrong data registration
		Low validation of data
Structured content published	Unstructured content in RDF	Encode web information system implementation error

3.2 Late Requirements

In this phase the tasks performed are *Develop Domain Model in OntoUML*, in which FrameWeb-LD prescribes the creation of a conceptual model of the domain elements of the system; *Develop Goal Model in iStar*, in which the goal model is created with the objective of identifying and modeling actors, goals, qualities, tasks, resources and other related elements for Linked Data systems; *Develop Risk Model in RiskML* in which, based on the identified risks, the risk model is created separated by categories (according to the tasks performed and goals to be achieved, for example, vocabulary adoption, data publication, data provenance, among others) and; finally, *Integrate Goal Model with Risk Model*, which has the objective of analyzing the impact relation of a risk event on a goal.

3.2.1 Develop Domain Model in OntoUML

The second phase of *GRALD* starts with the *Develop Domain Model in OntoUML* activity. Based on elicited system requirements and competency questions, inputs of this activity, the ontology is designed in OntoUML (GUIZZARDI, 2005), as discussed in Section 2.6.

According to Falbo (2014), important concepts and relations must be identified

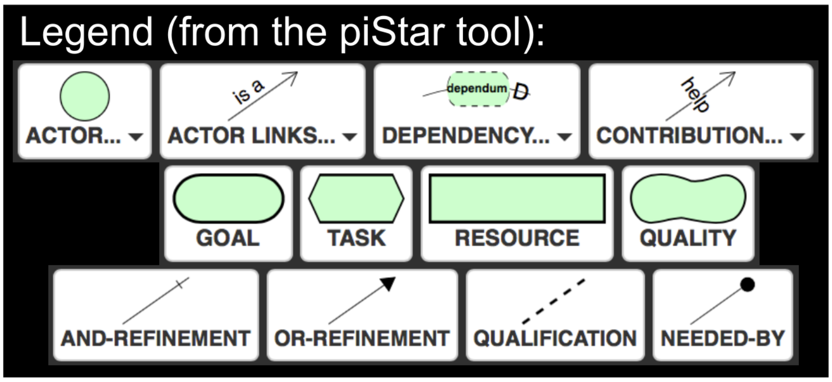

Figure 18 – Legend iStar 2.0 elements

and organized, and their analysis should be made based on a foundational ontology. The ontology engineer has the goal of identifying the main concepts and relations of the domain, the knowledge can be obtained through domain experts, books, international standards, reference models, etc.

According to Guizzardi (2005), the development of a domain model based on OntoUML aims to create a model with greater expressiveness of the domain, to establish a consensus between the experts and to obtain a shared conceptualization of the domain. The output of this activity are the Ontology Domain Model.

3.2.2 Develop Goal Model in iStar

The purpose of this activity is to model the goals (requirements) of the system, with a particular focus on publication of Linked Data. The inputs of this stage are the domain ontology model and the requirements specification document. Through goal modeling we can identify actors (stakeholders) and the relationship between them, goals to be achieved, tasks to be performed, resources to be employed, links between elements, etc.

Concepts related to GORE approaches and iStar were discussed in sections 2.2 and 2.3, respectively. In this section, we illustrate this step of the GRALD process with our running example, C2D. All goal models were built with the aid of the piStar tool.[3] Figure 18 shows the legend of the iStar 2.0 elements used.

Figure 19 shows the Strategic Dependency Diagram for C2D, depicting the actors involved and their interdependencies. The *Community* has the goal dependence *Keep info open for the community* on the *C2D* system. In turn C2D has two goal dependencies, *Obtain ontologies* and *Obtain RDF*, on the *Ontology Engineer* and the *Programmer* respectively, for the publication of interconnected data with FrameWeb-LD. The *Ontology Engineer* needs the knowledge about domain of the C2D system and, for this, she depends on the *Domain Expert*. Finally the *Programmer* needs the operational ontology (OWL) to

[3] <http://www.cin.ufpe.br/~jhcp/pistar/>).

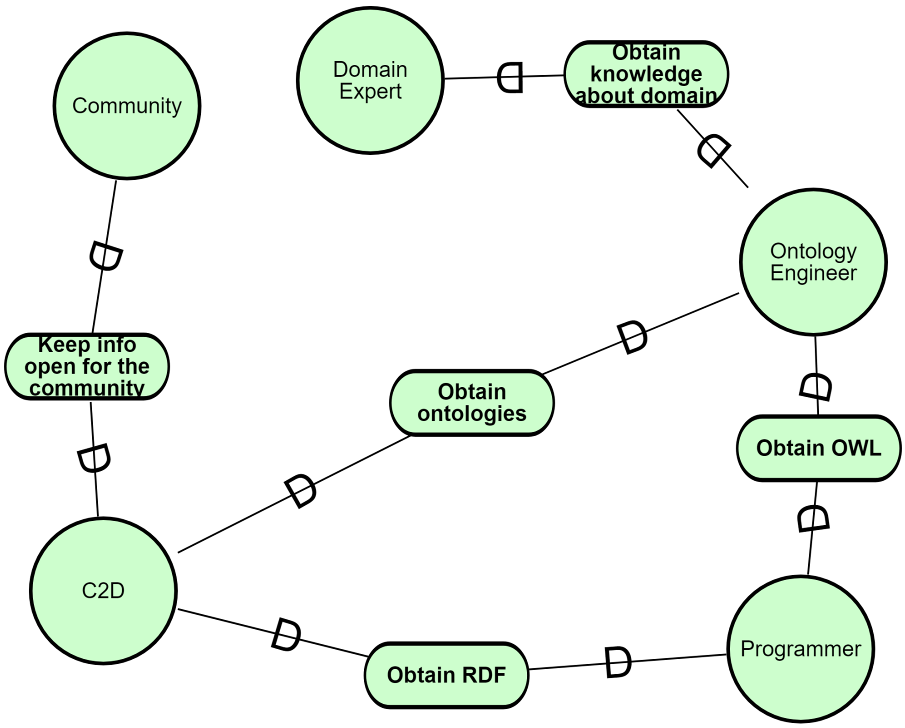

Figure 19 – iStar goal model SD (Strategic Dependency) view.

generate RDF for C2D.

After the strategic view is established, goal models are produced considering the point of view of each actor involved in the system, i.e., showing how they will provide the dependencies for which they are responsible. As with risk identification, some of these goal models may look generic and be (partially or fully) applicable in the development of other Linked Data Systems, which motivated us to propose the catalog of risks and goals, presented later in Section 3.4.

For instance, the actor *Domain Expert*, in Figure 20 has the objective of providing the knowledge to be modeled in the domain ontology by the Ontology Engineer (FALBO, 2014), in this case the application domain of the C2D system. She is responsible for the tasks *Elicit Requirements* and *Elaborate CQs*.

Figure 21 shows the goal model for the actors *Ontology Engineer/Ontology Designer*. These actors are suggested by Falbo (2014) and, in our case, for publication by FrameWeb-LD, they have the main goal *Ontologies Provided* and two sub-goals, the first being *Domain Model provided*. To achieve this goal they execute the task *Create Domain Model*, which helps *Establish a shared conceptualization*, *Greater expressiveness of the model* and *Establish consensus amid experts*; and performs the subtasks *Identify main concepts/ relations* using the resources *Competency Questions* and *Requirements Elicited*; *Organize main concepts/relations* and *Specify constraints/ inferences*, which uses *OCL languages*.

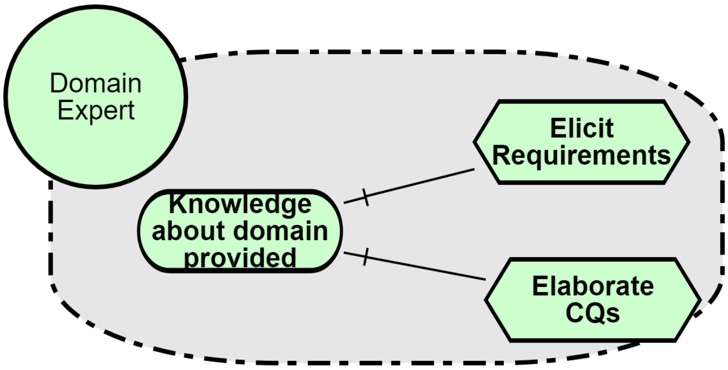

Figure 20 – iStar goal model for Domain Expert actor.

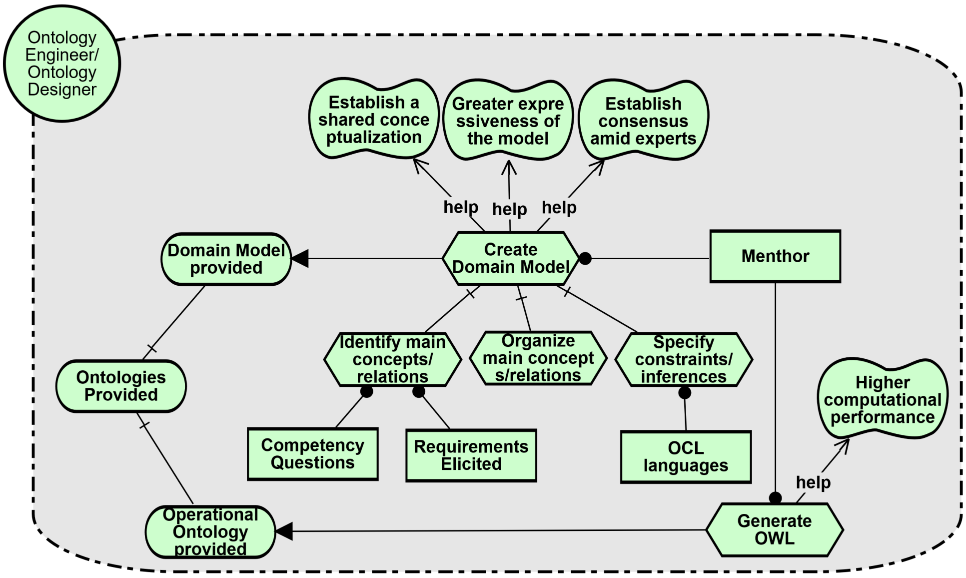

Figure 21 – iStar goal model for Ontology Engineer Actor.

According to Falbo (2014), important concepts and relations should be identified and organized by taxonomies. The concepts should be classified according to the types defined in OntoUML (kind, subkind, phase, role, category, rolemixin etc.), constraints and inferences must also be taken into account. The second sub-goal is *Operational Ontology provided*: the task *Generate OWL* is performed and helps to *Higher computational performance*, because this type of ontology is processed by machines. The resouce *Menthor* is used to *Create Domain Ontology* and *Generate OWL*.

In Figure 22, the actor *Programmer* has the main goal *RDF provided*, accomplished by the main task *Generate RDF*, using the resource *D2RQ* (in order to automate the generation of RDF from the database). In turn, the main task depends on two other tasks: *Chosse vocabulary* and *Encode Web Information System*, which respectively use the resources *LOV* and *ReMat*. *LOV*[4] is a search engine for Linked Data vocabularies and *ReMat* is a tool proposed by Celino et al. (2016) for relating the OWL file to external

[4] http://lov.okfn.org/dataset/lov/

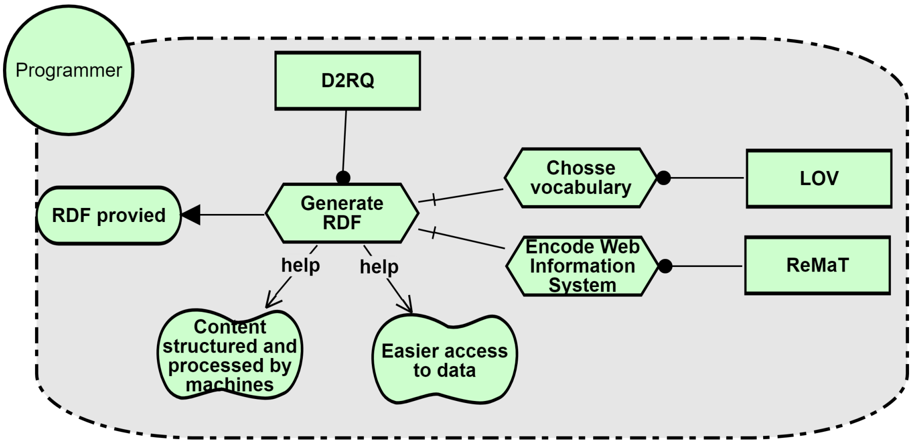

Figure 22 – iStar goal model for Programmer Actor.

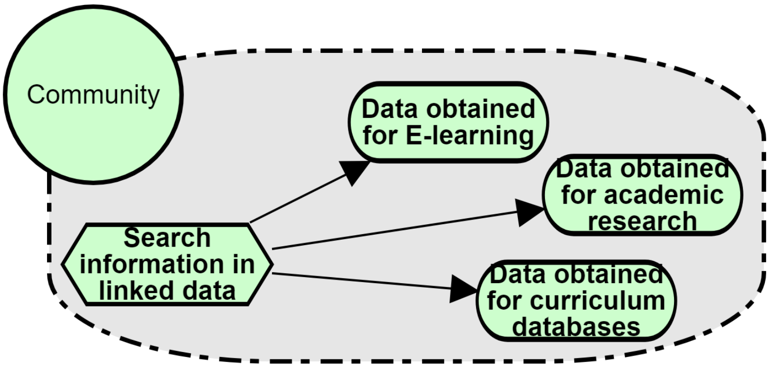

Figure 23 – iStar goal model for Community Actor.

vocabularies. The main task helps *Content structured and processed by machines* and *Easier access to data.*

The actor *Community*, in Figure 23, represents the academic community, composed by students, professors (researchers), staff, etc. As such, the community has the goals *Data obtained for E-learning*, *Data obtained for academic research* and *Data obtained for curriculum databases*, accomplished by the task *Search information in Linked Data*. These are goals that apply particularly in the case of C2D.

Finally, Figure 24 shows the goal model for the C2D system itself, deployed and maintained in our university. The central goal for C2D, therefore, is *Data Published in Linked Data*, divided in subgoals, according to the data that will be published: *Scores*, *Venues*, *Publications* and *Researchers*. The goal *Users not published in Linked Data* represents the fact that user data should not be published. The data is registered in the system by the tasks *Calculate researcher score*, *Manage venues*, *Manage publications*, *Manage researchers* and *Manage and authenticate users.*

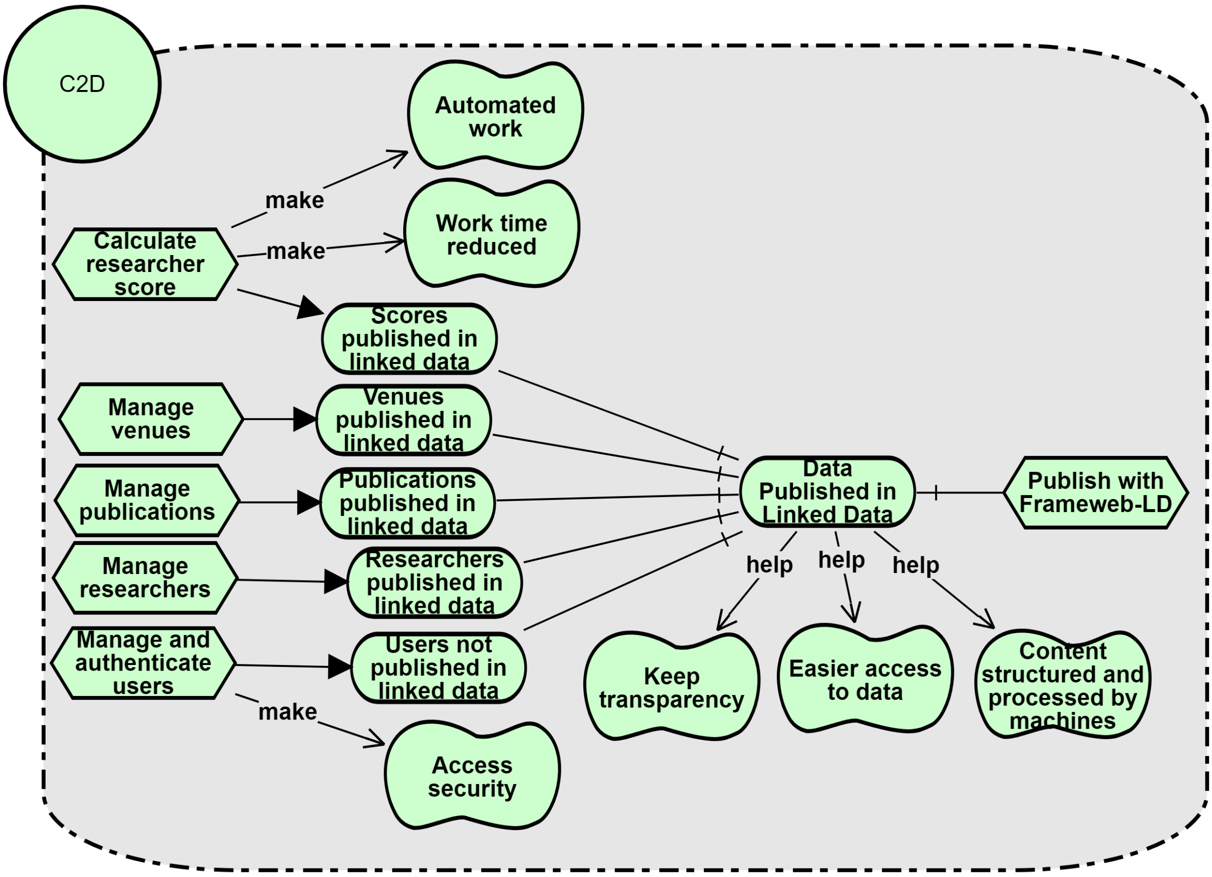

Figure 24 – iStar goal model for C2D Actor.

About the *qualities* of the system, the main goal *Data Published in Linked Data* helps C2D to *Keep transparency* because the data on researcher accreditation are open for the community to search; *Content structured and processed by machines* and *Easier access to data* are helped because the data is published in RDF format, allowing the possibility of a computational agent to process it. The task *Calculate researcher score* makes the *Automated work* and *Work time reduced* sofgoals, eliminating the manual work of calculating the scores, usually conducted by members of the program (i.e. researchers, with a doctoral degree) and considerably reducing the time taken to generate these scores for all researchers of the program. The task *Manage and authenticate users* makes *Access security*, ensuring access control to the system. The data is published by the FrameWeb-LD (CELINO et al., 2016) systematic approach, discussed in Section 2.6.

Thus, the outputs of this stage are the iStar goal models. Their elements will be further described and explained in Section 3.4.

3.2.3 Develop Risk Model in RiskML

Based on the results of the *Identify Risks* (inputs of this activity) phase (cf. Section 3.1.2), risk models based on RiskML language are created in this activity. The situations and events of risks, as well as the new goals are modeled, with the purpose of

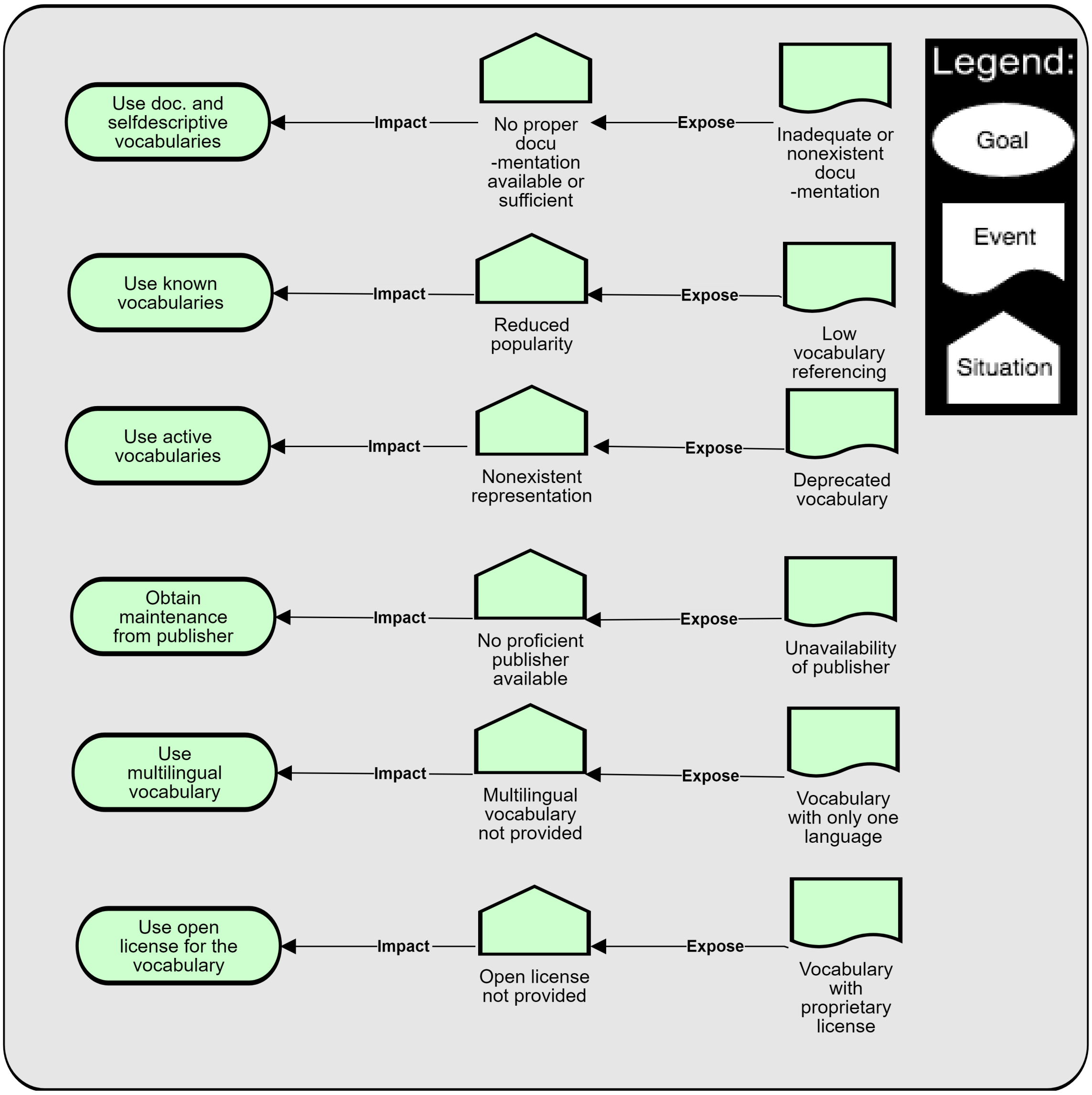

Figure 25 – Vocabulary adoption Risk Model in RiskML language.

demonstrating the impact of the events on the goals.

Figure 25 shows the risk model related to vocabulary adoption in C2D. For example, the goal *Use documented and self-descriptive vocabularies* is impacted by risk event *No proper documentation suficient or available* exposed by risk situation *Inadequate or nonexistent documentation*. Also, the goal *Use active vocabularies* is impacted by risk event *Non-existent representation* exposed by risk situation *Deprecated vocabulary*, and so on.

Figure 26 shows the risk model related to data publication. For instance, the goal *Use cool URI* is impacted by risk event *Not provide URI in accordance the best practices* exposed by risk situation *URI in non-compliance with best practices* and the goal *Structured content published* is impacted by the risk event *Unstructured content in RDF* exposed by the situation *Encode web information system implementation error*.

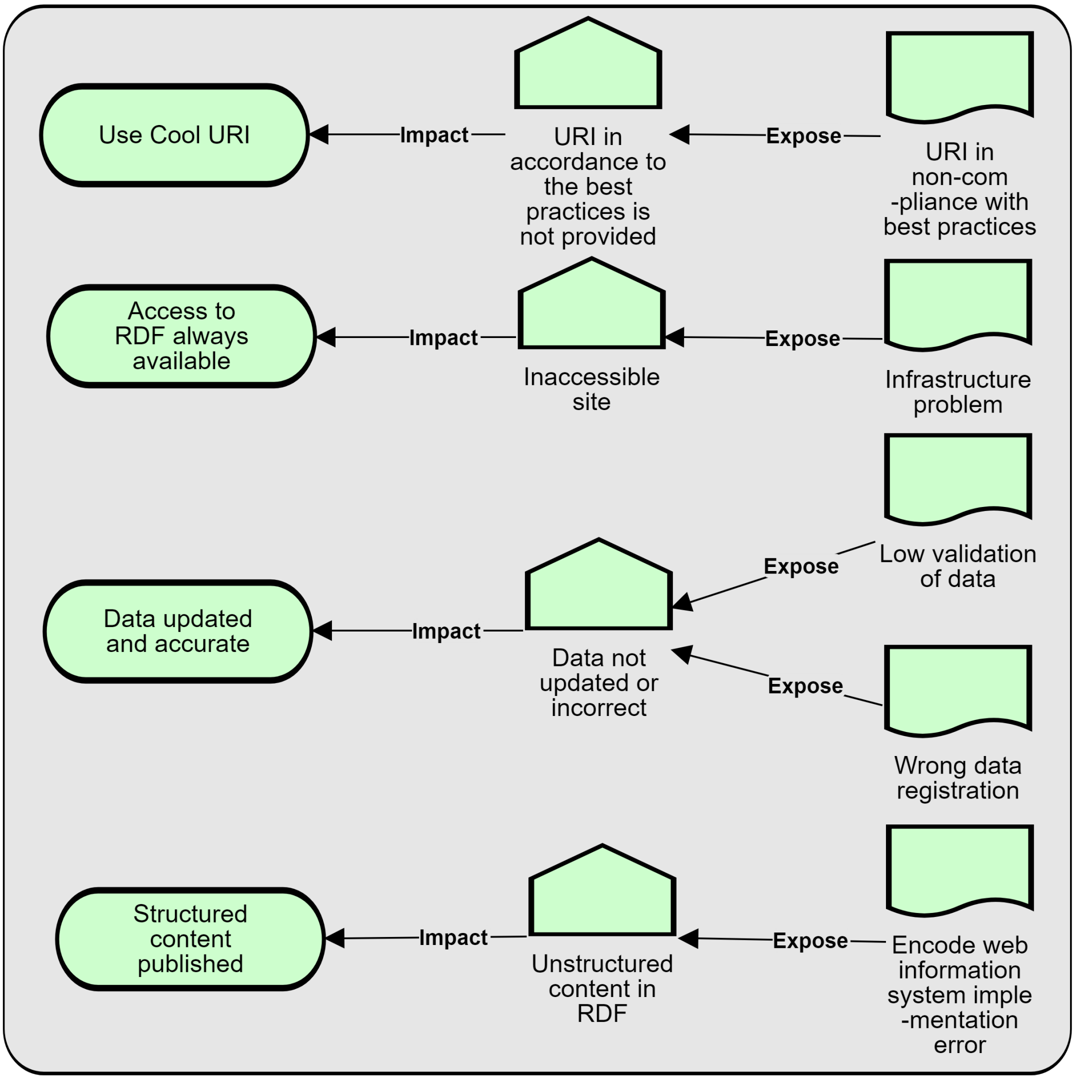

Figure 26 – Data publication Risk Model in RiskML language.

Other risks related to the creation and maintenance of vocabulary and ontology, dataset selection, trust and proof of information, publishing data and traditional Web risks can be identified and modeled in separate models. The description and explanation about the elements of the risk models are available in Section 3.4. The outputs of this stage are the risk models created with RiskML language.

3.2.4 Integrate Goal Model with Risk Model

Based on RISCOSS, the last activity of this phase is *Integrate Goal Model with Risk Model*, aligning goals and risks. To this end, goals that were elicited during the construction of the RiskML model are added to the iStar model and are associated with existing goal model elements. At this point, elements from both models can be maintained, added or discarded in order to produce an integrated model. The inputs of this stage are the goal and risk models.

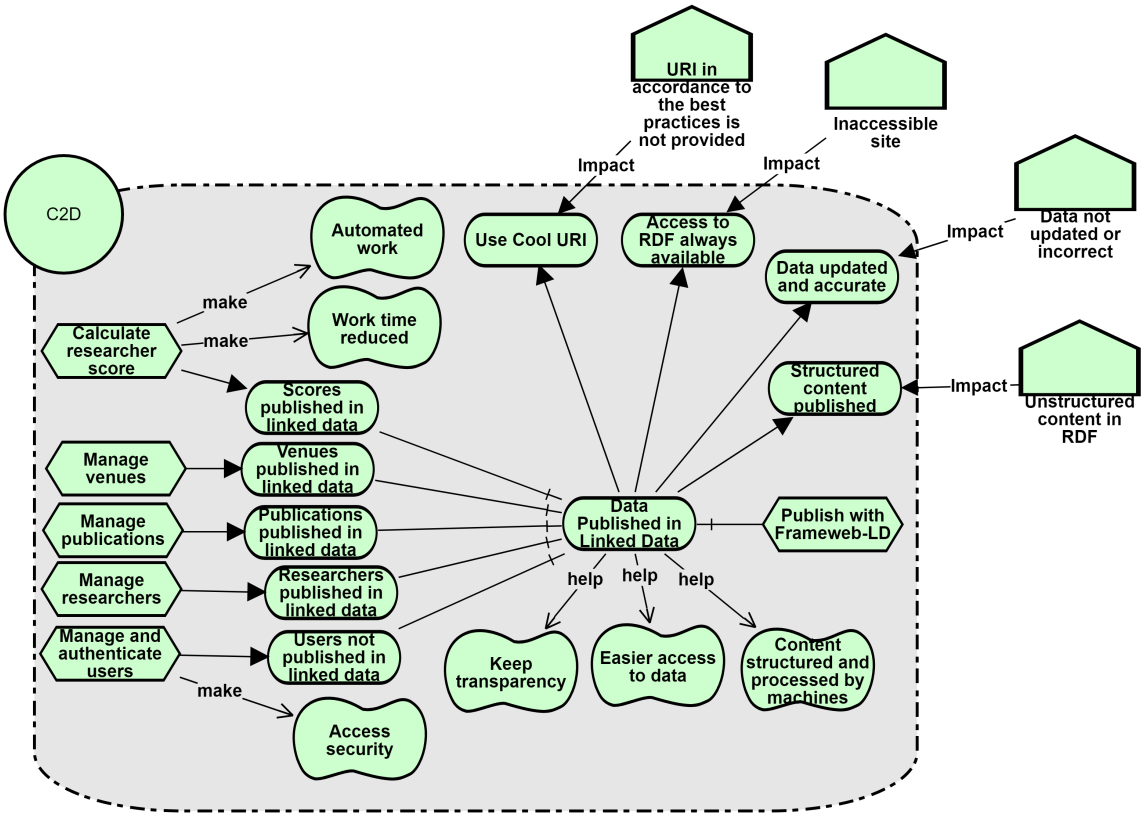

Figure 27 – Goal model for Linked Data implementation, related to data publication, connected to RiskML risk events.

In Figure 27, new goals related to data publication are added to the model: *Use cool URI*, *Access to RDF always available* and *Data updated and accurate*, impacted by the risks events *No URI in accordance with the best practices*, *Inaccessible site* and *Data not updated or incorrect*, respectively. These goals were added with the objective of creating the integrated model, to demonstrate the impact relation.

Figure 28 shows the integrated model related to choice of vocabularies in Linked Data systems development. Regarding the task Choose vocabulary, new goals are added to the model, such as *Use active vocabulary*, impacted by the risk event *Non-existent representation*; *Use know vocabularies*, impacted by the risk event *Reduced popularity*; *Use documented vocabularies* impacted by the risk event *No proper documentation available*, among others.

Once the models are integrated, risk analysis can be performed as per (COSTAL et al., 2015). The impact relation between a risk and a goal represents a negative effect when the event is likely and significant, increasing the evidence that the goal is not achieved. Such evidence is then propagated through the goal graph calculating, for each intentional element, if it is totally/partially satisfied/denied. We are then able to see how risks affect the strategic/high-level goals of each of the involved actors and prioritize our risk mitigation efforts based on this analysis.

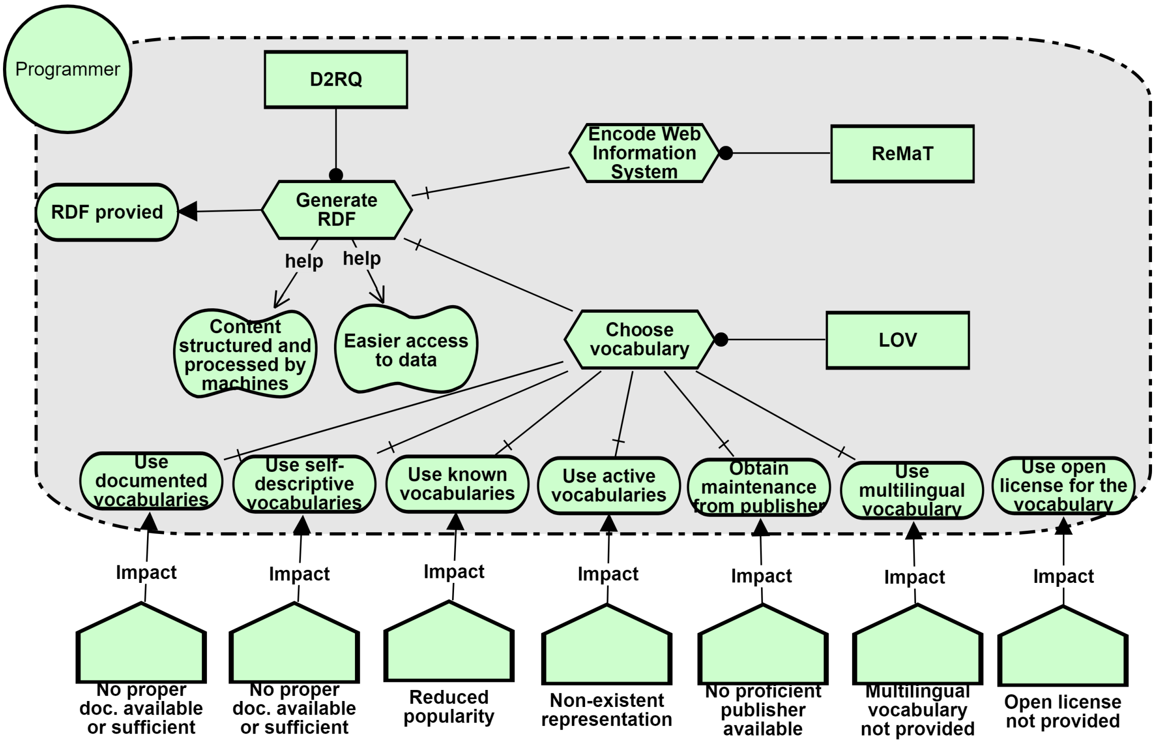

Figure 28 – Goal model for Linked Data implementation, related to choice of tools and vocabularies, connected to RiskML risk events.

The figures concentrate on goals related to the publication of Linked Data only. We can thus see how risk propagation can impact the main goal *Data Published in Linked Data*, which interests the actors C2D and Community. For example, the risk event *Non-existent representation* can impact the goal *Active Vocabularies Used* and, in this case, other vocabularies will have to be chosen, incurring in system maintenance. Further, the risk event *Reduced Popularity* impacts the goal *Know vocabularies used* if the adopted vocabulary is not well referenced by other datasets in the Linked Data community. The total or partial dissatisfaction of the goals *Use documented and self-descriptive vocabularies* and *Obtain maintenance from publisher* can hinder the process of vocabulary adoption and maintenance. In these cases, a benefit of risk analysis is to help in the choice of vocabularies. The outputs of this phase are the integrated models.

3.2.5 Development Tool

In this work, we were particularly concerned about the creation and maintenance of the models. As already discussed, goal models are created in the iStar 2.0 language, and the risk model using RiskML language.

For goal modeling, we use the piStar tool, provided by the researcher João Pimentel from Federal University of Pernambuco - UFPE, under the MIT license. The piStar tool

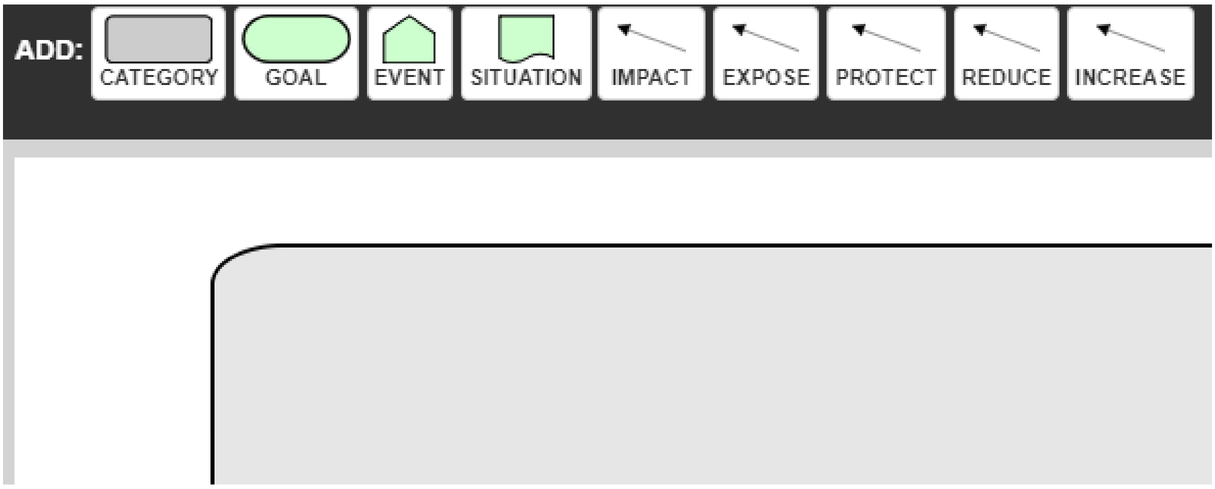

Figure 29 – piStar adapted for RiskML model.

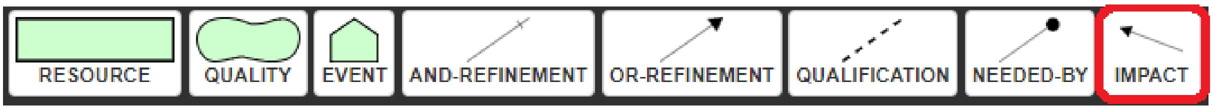

Figure 30 – piStar adapted for Integrated model.

allows you to draw iStar goal models, serialize them in the JSON format and to export them as images in SVG or PNG formats (PIMENTEL; CASTRO, 2018).

For the creation of risk and integrated (risk and goals) models, we extended the piStar tool in order for it to support elements related to the RiksML language (Situation, Event), as well as its relations (Increase, Reduce, Protect, Expose and Impact), as shown in Figure 29. Also, another extension, shown in Figure 30, was made to create the integrated models, the impact of a risk event on a goal, illustrated, for instance, in Figure 27. The source code for these extensions is available in a public repository: <https://github.com/nemo-ufes/FrameWeb-GRALD/tree/master/Tool/GraldPortal>;

3.3 Design

Based on tasks *Elicit Requirements*, *Develop Domain Model in OntoUML*, *Develop Goal Model in iStar* and *Integrate Goal Model with Risk Model*, design begins with the *Search for Candidate Vocabularies* for Linked Data publication. The activities in the previous phases help identify existing classes and relations and, based on them, we can search for vocabularies. After choosing the vocabularies, the task *Create FrameWeb-LD Entity model* is performed.

[5] <http://watson.kmi.open.ac.uk/WatsonWUI/>
[6] <http://prefix.cc/>

3.3.1 Search for Candidate Vocabularies

For this activity, Hyland, Atemezing and Villazón-Terrazas (2014) suggest Linked Data search engines such as Linked Open Vocabularies (LOV), Watson,[5] Prefix.cc,[6] Swoogle[7] or Bioportal (biological domain).[8]

According to these authors, in the process of choosing a vocabulary we must take into account if the vocabularies are published by a trusted group or organization, if the vocabularies have permanent URIs, confirm a version control policy, choose documented vocabularies, choose self descriptive vocabularies, choose vocabularies described in more than one language, choose vocabularies used by other data sets and choose vocabularies that are available for access for a long or infinite time. These recommendation form a checklist developers should go through in order to determine the quality of each candidate vocabulary.

Regarding the choice of vocabularies, for our running example, we used with more emphasis the search engine Linked Open Vocabularies (LOV). To search for vocabulary classes for the *Researcher*, *Publication* and *Venue* domain classes, we searched the categories (tags) *People*, *Catalogs* and *Academy*. Analyzing results using the aforementioned recommendation checklist resulted in the choice of new vocabularies for C2D, namely *Schema.org*, *DBPedia*, *Bio*, *Bibtex* and *Bibo*. Analyzing links between vocabularies also helped in the discovery of new vocabularies to consider (with respect to what had already been chosen by Celino et al. (2016)).

The check-list used in this process is shown in Table 4. Vocabulary attributes are presented in different rows, whereas the columns indicate if the vocabularies being checked meet the criteria (represented by a checkmark: ✓), do not meet the criteria (represented by an $\times$), or partially meet the criteria (represented by a plus/minus sign: $\pm$). To check each attribute, the data presented by LOV was analyzed, as well as the vocabularies' own documentation and their OWL schema. In the case of C2D, all attributes were considered to have the same weight, but different weights can be assigned depending on the system being developed.

Linked Data search engines, such as LOV, provide vocabulary information such as label, URI, namespace, description, creator, publisher, comment and language. Also, information such as vocabulary version history is important to measure the reliability of vocabulary regarding the level of updates that may represent the addition of new classes, properties and deprecated classes. Incoming links represents the popularity of vocabulary because it means that other projects are referencing it. Below, we further describe how each item of the checklist can be verified:

7 <http://swoogle.umbc.edu/2006/>
8 <http://bioportal.bioontology.org/>

Table 4 – Vocabulary checklist for C2D.

#	**Attributes**	**Dbo**	**Schema**	**Bibo**	**Bio**	**Bibtex**
1	Published by a trusted group or organization	✓	✓	✓	✓	✓
2	Have permanent URIs	✓	✓	✓	✓	✓
3	Version control policy	✓	✓	✓	✓	✓
4	Documented vocabularies	✓	✓	✓	✓	✓
5	Self descriptive vocabularies	✓	✓	✓	✓	✓
6	Described in more than one language	✓	×	×	×	×
7	Used by other data sets	✓	✓	✓	✓	±
8	Available for access for a long or infinite time	✓	✓	✓	✓	✓
9	Related to the domain	×	×	✓	×	✓

- Item 1: check if the vocabularies have at least one creator, URI and namespace;
- Item 2: check if the URI is stable;
- Item 3: check if the vocabulary uses any sort of versioning system, e.g., are there previous versions with different numbering?
- Item 4: check if the vocabularies have websites with their respective documentation;
- Item 5: check the vocabulary OWL schema for triples that describe its classes and properties (e.g., comments or labels);
- Item 6: check the vocabulary OWL schema for strings in more than one language, considering your target audience (in our example, Dbo was the only vocabulary that met this criterion);
- Item 7: check if the vocabulary has a substantial amount of incoming links (in our example, LOV indicated Bibtex had only a single incoming link, therefore we consider that it partially met this criterion);
- Item 8: check for how long the vocabulary has been maintained and if they are published in a stable domain, verifying its documentation;
- Item 9: check if the vocabulary is related to the same application domain, which for C2D is the case of Bibtex and Bibo.

The above checklist is, of course, not exhaustive and could be improved with further vocabularies and/or desired attributes to check, depending on the availability of resources involved in the software development project. The vocabularies, tables and check-list above are the outputs of this activity.

3.3.2 Create FrameWeb-LD Entity model

Once the vocabularies, which are the inputs for this activity, are chosen, we move on to *Create FrameWeb-LD Entity Model*. In this activity, we build an Entity Model as proposed by Celino et al. (2016), by adding Linked Data mapping annotations to the domain model, based on FrameWeb-LD meta-model, to the vocabulary chosen in the previous activity.

Figure 31 represents the model built for C2D, which is based on the model exemplified in (CELINO et al., 2016) (shown in Figure 16, Section 2.6), with new vocabulary classes added by the process suggested by *GRALD*, which are filled in blue. For instance, for the domain class `Researcher`, vocabularies `schema:Person`[9] and `dbo:Person`[10] were added; for `Publication`, `bibo:Article`[11] and `bibtex:Article`;[12] and for `Venue`, `schema:Organization`[13] and `bio:Organization.`[14], The Entity Model is the output os this activity.

3.4 Linked Data Risks and Goals Catalog

Based on the iStar metamodel, the RISCOSS approach and FrameWeb-LD, in this work we suggest a Catalog for Linked Data implementation with goals, tasks, resources, qualities and risks events and situations. The objective of this catalog is to provide knowledge for the construction of models of other systems, as well as to support in the evaluation of our proposal. For the creation of the catalog, references such as (HYLAND; ATEMEZING; VILLAZóN-TERRAZAS, 2014), W3C (2017) and Falbo (2014) were taken into account.

- **Goals - Data Publication**

  Access to RDF always available: access to RDF always available to be processed by machines.

  Data Published in Linked Data: represents the main goal of the system regarding data publication.

  Keep info open for the community: goal to be achieved by a system regarding the need to keep info open to a particular community.

  Obtain RDF: this goal represents the need for a system to obtain the RDF for the publication of linked data.

[9] http://schema.org/Person
[10] http://dbpedia.org/ontology/Person
[11] http://purl.org/ontology/bibo/
[12] http://zeitkunst.org/bibtex/0.2/bibtex.owl
[13] http://schema.org/Organization
[14] http://vocab.org/bio/

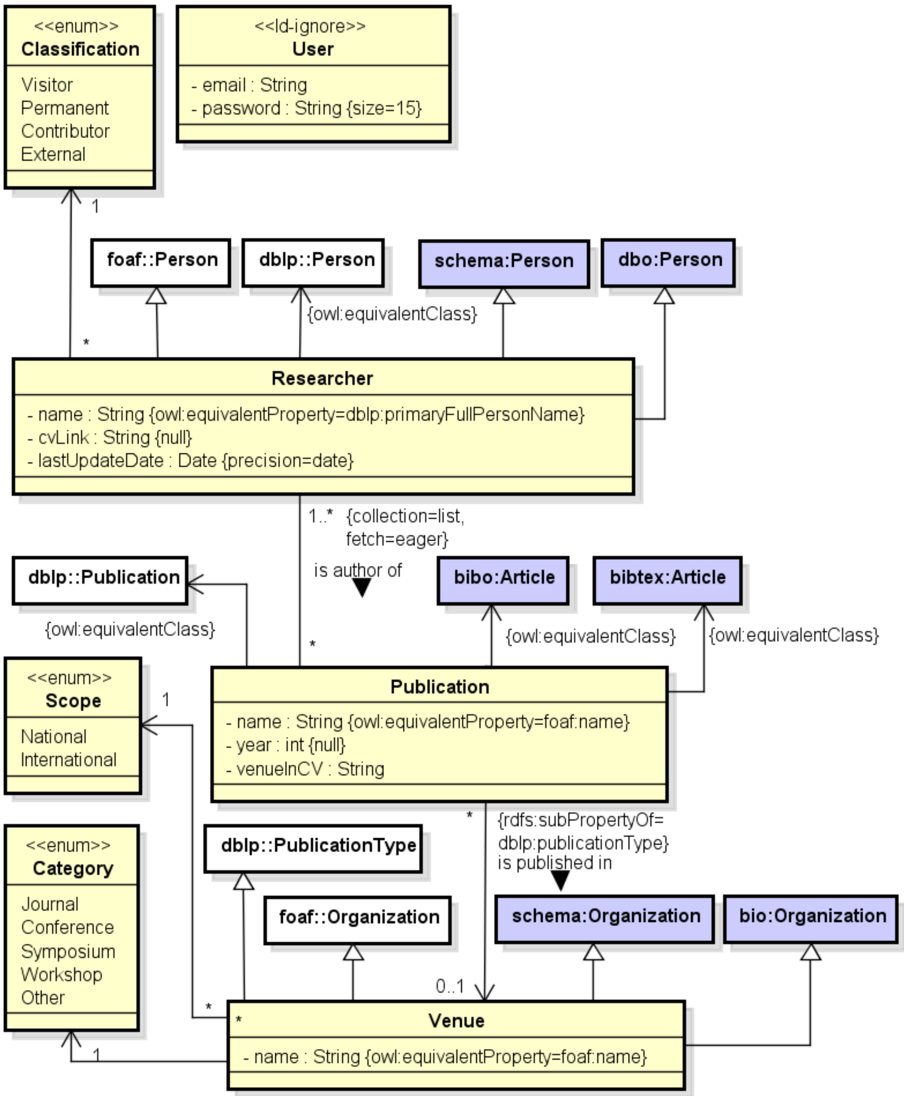

Figure 31 – The FrameWeb-LD Entity Model for C2D with newly added vocabularies during the GRALD process.

Use cool URIs: use adequate URIs for data publication according the *URI Design Principles* and *URI Construction* suggested by W3C (HYLAND; ATEMEZING; VILLAZóN-TERRAZAS, 2014).

- **Goals - Vocabulary Adoption**

  Use active vocabularies: this goal refers to the use of active and non-depreciated vocabularies in the representation of the content published in Linked Data.

  Use documented and self-descriptive vocabularies: this goal refers to the use of vocabulary documents, as well as with description of your classes and properties (e.g., comments or labels).

  Use known vocabularies: this goal refers to the vocabulary which has a substantial amount of incoming links.

Obtain maintenance from publisher: this goal refers to the use of vocabularies published by an organization that supports and updates vocabulary, such as adding / removing new classes and properties.

Use multilingual vocabulary: this goal refers to the use of vocabularies with more than one language, among a relevant set of languages in your context.

Use open license for the vocabulary: this goal refers to the use of vocabularies with open usage license.

- **Goals - Ontologies and RDF**

  Domain Model provided: the ontology engineer's goal to be achieved to provide the system Domain Ontology.

  Knowledge about domain provided: represents the main goal of a *Domain Expert*, aiming to assist the *Ontology Engineer* to create the Domain Ontology.

  Obtain knowledge about domain: this goal represents the need for the *Ontology Engineer* obtain the knowledge about the domain of the system on which the *Domain Ontology* will be created.

  Obtain ontologies: this goal refers to the need for an actor to obtain ontologies, be it Domain or Operational Ontology.

  Obtain OWL: this goal refers to the need for an actor to obtain Operational Ontology.

  Ontologies Provided: refers to the main goal of the *Ontology Engineer*, to provide the Domain or Operational Ontology.

  Operational Ontology provided: goal to be reached to provide the Operational Ontology for the publication of data in Linked Data.

  RDF provided: goal to be reached to provide the RDF for the publication of data in Linked Data.

- **Goals - Search data in Linked Data datasets**

  Data obtained from specific domain: this goal refers to data obtained from specific domains, in datasets such as Social Networking, Media and Publications. These data can be obtained through a SPARQL Endpoint, for example.

  Registration optimized data: data obtained through dataset can optimize the task of data register in the system.

  Reliable obtained data: this goal refers to the obtaining of reliable data in datasets, for to register them in the system, being necessary a validation of this data

Get updated data: this goal refers to the obtaining of updated data in data sets, for to register them in the system, being necessary the validation of this data.

- **Goals - Data provenance**

  Provide details about the data origin: this goal refers to provide details about the data origin. According to the W3C (2017), properties such as *dct:creator*, *dct:publisher* and *dct:issued*, in the *Data Catalog Vocabulary* (DCAT)[15] can be used to provide information about the data origin.

  Ensure the provenance of the data: this goal refers to provide provenance about the data published.

  Provide credibility and data integrity: this goal refers to provide credibility and data integrity about the data published.

- **Goals - SPARQL Endpoint provided**

  Provide SPARQL Endpoint available: this goal refers to provide an SPARQL Endpoint available, where data can be queried/consumed by other agents.

  Provide up-to-date data: this goal refers to provide updated data for an agent consumption.

  Provide reliable data: this goal refers to provide reliable data for an agent consumption.

- **Goals - RDF license**

  Provide data license information: this goal refers to provide details about the data license. According to the W3C (2017), some vocabularies such as *dct:license*, *cc:license*, *schema:license* and *xhtml:license* can be used.

  Provide machine-readable license : this goal refers to provide RDF machine-readable license for an agent.

  Provide human-readable license: this goal refers to provide human-readable license for an agent.

- **Qualities - Data Publication**

  Easier access to data: quality to be achieved with the publication of interconnected data for easy access in published content.

  Keep transparency: quality to be achieved with the publication of interconnected data for transparency in processes related to, for example, a public sector.

[15] <https://www.w3.org/TR/vocab-dcat/>

Content structured and processed by machines: quality to be achieved with the publication of interconnected data for data processed by machines on LOD Cloud.

- **Qualities - Ontologies and RDF**

Establish consensus amid experts: the ontological model helps to establish a consensus among domain experts.

Establish a shared conceptualization amid experts: the ontological model helps to establish a shared conceptualization among domain experts.

Greater expressiveness of the model : the creation of the Domain Ontology with OntoUML helps to greater expressiveness of the model.

Higher computational performance: the task Generate OWL helps to higher computational performance, because this type of ontology is processed by machines.

- **Tasks - Data Publication**

Publish with FrameWeb-LD: represents the main task to data publication, following the FrameWeb-LD guidelines or other way of publication.

Store data in triple store: represents the task to store data in triple store, for SPARQL queries.

- **Tasks - Ontologies and RDF**

Create Domain Model: main tasks performed by the ontology engineer, this taks helps to establish a shared conceptualization, greater expressiveness of the model and establish consensus amid experts.

Choose vocabulary : in this task the external vocabularies are selected for the association with the Operational Ontology (OWL).

Elaborate CQ's: task performed by a *Domain Expert* or *Ontology User* with a objective of elicit Functional Requirements to be represented by the ontology, questions that the ontology should be able to answer (FALBO, 2014).

Elicit requirements: task performed by a *Domain Expert* or *Ontology User* with a objective of elicit Functional and Non-Functional Requirements.

Encode Web Information System : in this task the external vocabularies are associated with the OWL schema.

Generate RDF: in this task the RDF is generated, with the vocabularies associated, for the publication of data.

Generate OWL: in this task, from the Domain Ontology, the Operational

Ontology is generated.

Identify main concepts/relations: according to Falbo (2014), in order to create the Domain Model the relevant concepts and relation should be identified.

Organize main concepts/relations: according to Falbo (2014), the relevant concepts should by organized by taxonomies. The concepts should be classified according to the types defined in OntoUML (kind, subkind, phase, role, category, rolemixin, etc.).

Search information in Linked Data: this task is performed by a stakeholder which has the need to seek some information in the linked data cloud.

Specify constraints/inferences: according to Falbo (2014) constraints and inference rules can be applied on the Domain Model

- **Resources**

Design Domain Ontology tools: tool used to create a Domain Ontology and Generate OWL (Operational Ontology), e.g., OLED[16] and Menthor.[17]

Competency Questions (CQ's): questions elaborated and answered with the objective of helping to create the Domain Ontology (FALBO, 2014).

Requirements Elicited: the elicited requirements help an *Ontology Engineer* to create the Domain Ontology.

OCL languages: are used to specify constraints and inferences rules in Domain Model.

Linked Data Search Engines: are used to search data sets and vocabularies. Hyland, Atemezing and Villazón-Terrazas (2014) suggest Linked Open Vocabularies (LOV), Watson,[18] Prefix.cc,[19] Swoogle[20] or Bioportal (biological domain).[21]

Encode web information System tools: these tools are used to relate external vocabularis with the OWL for RDF generation, e.g., Jena,[22] RDF.Net[23] and ReMaT, proposed by Celino et al. (2016).

Triple Store: database for triple storage and retrieval, like Stardog, Virtuoso and others.

[16] https://nemo.inf.ufes.br/projects/oled/
[17] http://www.menthor.net
[18] <http://watson.kmi.open.ac.uk/WatsonWUI/>
[19] <http://prefix.cc/>
[20] <http://swoogle.umbc.edu/2006/>
[21] <http://bioportal.bioontology.org/>.
[22] <https://jena.apache.org/>
[23] <http://www.dotnetrdf.org/>

- **Risk Situations - Vocabulary Adoption**

  Inadequate or nonexistent documentation: the situation of lack of documentation or inadequate documentation, exposes the risk event *No proper documentation available or sufficient.*

  Low vocabulary referencing: the situation in which the vocabulary is poorly referenced by others, exposes the risk event *Reduced popularity.*

  Deprecated vocabulary: this situation exposes the risk event *Nonexistent representation.*

  Unavailability of publisher this situation exposes the risk event *No proficient publisher available.*

  Vocabulary with only one language: vocabulary described in only one language, exposes the risk event *Multilingual vocabulary not provided.*

  Vocabulary with proprietary license: this situation expose the risk event *Open license not provided.*

- **Risk Situations - Data Publication**

  URI in noncompliance with best practices: this situation exposes the risk event *URI in accordance to the best practices is not provided*, suggested by (HYLAND; ATEMEZING; VILLAZóN-TERRAZAS, 2014).

  Infrastructure problem: Some infrastructure problem, for example Internet or server link crash, exposes the risk event *Inaccessible site.*

  Low validation of data: the situation in which the data registered are poorly validated, exposes the risk event *Data not updated or incorrect.*

  Wrong data registration: the situation in which the data is incorrectly entered, exposes the risk event *Data not updated or incorrect.*

  Encode web information system implementation error: error in the implementation of an RDF, exposes the risk event *Unstructured content in RDF.*

- **Risk Situations - Search data in Linked Data datasets**

  Infrastructure or network problem: Some infrastructure or network problem, for example Internet or server link crash, exposes the risk event *SPARQL Endpoint not available.*

  Low validation of data: the situation in which the data searched/ consumed are poorly validated, exposes the risk event *Data not updated or incorrect.*

- **Risk Situation - Data provenance**

RDF generated without the properties of provenances: this risk situation exposes the risk event Data published without provenance metadata, where RDF does not contain information related to data provenance, such as *dct:creator*, *dct:publisher* and *dct:issued*.

- **Risk Situation - SPARQL Endpoint provided**

  Infrastructure or network problem: Some infrastructure or network problem, for example Internet or server link crash, exposes the risk event *SPARQL Endpoint not available*.

  Low validation of data: the situation in which the data provided are poorly validated, exposes the risk event *Data not updated or incorrect*.

- **Risk Situations - RDF License**

  RDF generated without license properties: the situation in which RDF does not contain information about the license, exposes the risk event *License data not provided*.

  Page with license not provided: the situation in which the page concerning the published data does not contain information about the license, exposes the risk event *License data not provided*.

- **Risk Events - Vocabulary Adoption**

  No proper documentation available or sufficient: this risk event impacts the goal *Use documented and self-descriptive vocabularies*, because the documentation is very important for the choose of vocabularies, helps to understand classes properties and relationships.

  Reduced popularity: this risk event impact the goal *Use known vocabularies*, because the amount of incoming links is important for the choice of vocabularies.

  Non-existent representation: this risk event impacts the goal *Use active vocabularies*, in the case of, for example, deprecated vocabulary.

  No proficient publisher available: this risk event impacts the goal *Obtain maintenance from publisher* because it is very important the publisher of the vocabulary supports and updates the vocabulary.

  Multilingual vocabulary not provided: this risk event impacts the goal *Use multilingual vocabulary*, because the internationalization of the vocabulary should by considered.

  Open license not provided: this risk event impacts the goal *Use open license for the vocabulary*, because the use of open license can be considered.

- **Risk Events - Data Publication**

  URI in accordance to the best practices is not provided: this risk event impacts the goal *Use Cool URI*, because the URI is not compliant with best practices.

  Inaccessible site: this risk event impacts the goal *Access to RDF always available*, in the case of the occurrence of an infrastructure problem.

  Data not updated or incorrect: this risk event impacts the goal *Data updated and accurate* in the case of the wrong data registration or low validation of data.

  Unstructured content in RDF: this risk event impacts the goal *Structured content published*, in the case of encode Web information system implementation error.

- **Risk Events - Search data in Linked Data datasets**

  SPARQL Endpoint not available: this risk event represents the SPARQL Endpoint is unavailable for data querying, impacts the goals and *Data obtained from specific domain* and *Registration optimized data*.

  Data not updated or incorrect: this risk event impacts the goals *Reliable obtained data* and *Get updated data*.

- **Risk Event - Data provenance**

  Data published without provenance metadata: this risk event impacts the goals *Provide details about the data origin*, *Ensure the provenance of the data* and *Provide credibility and data integrity*, related to data published in Linked Data.

- **Risk Events - SPARQL Endpoint provided**

  SPARQL Endpoint not available: this risk event represents the SPARQL Endpoint is unavailable for data querying by an agent, impacts the goal *Provide SPARQL Endpoint available*.

  Data not updated or incorrect: this risk event impacts the goals *Provide reliable data* and *Provide to up-to-date data*.

- **Risk Event - RDF License**

  License data not provided: this risk event impacts the goals *Provide data license information*, *Provide machine-readable license* and *Provide human-readable license*, where the published data does not contain license information.

A first version of this catalog has been published at <https://github.com/nemo-ufes/FrameWeb-GRALD/wiki/A-Catalog-of-Goals-and-Risks-for-Linked-Data-Systems-Development> and should be periodically updated. This, however, is left for future work.

3.5 Conclusion of this Chapter

In this chapter the proposal of this work, named GRALD, was presented. GRALD seeks to establish a synergy between two approaches, RISCOSS and FrameWeb-LD. In the Early Requirements phase the tasks performed are *Elicit Requirements*, suggested by Celino et al. (2016), and *Identify Risks*.

In the second phase, Late Requirements, the task Develop Domain Model in OntoUML, suggested by Celino et al. (2016), takes place. Then, based on RISCOSS, the tasks Develop Goal Model in iStar, Develop Risk Model in RiskML and Integrate Goal Model with Risk Model follow. In the third phase, Design, candidate vocabularies are searched for the creation of the FrameWeb-LD Entity Model. Some benefits of our proposal are:

- Uses a GORE approach, iStar, to modeling requirements goal for Linked Data Systems;
- Prescribes risk identification related to Linked Data Systems;
- Prescribes risk modeling using RiskML language;
- Integrates Goal and Risk Model, with the objective of analyzing the impact of the risk event on stakeholder goals;
- Provides tools for goal modeling with iStar 2.0 and risk modeling based on RiskML language;
- Suggests a catalog of elements related on Goals and Risks for Linked Data system implementation.

We believe that the steps illustrated in this chapter help in satisfying our main objective of supporting developers of information systems on the Web in the analysis of goals and risks in the publication of Linked Data by these systems and in the choice of appropriate vocabularies.

4 Evaluation

The evaluation of this proposal was conducted by the author of this dissertation and the undergraduate student in Computer Science course, Allan Araujo Silva (SILVA, 2017), using Web-based Information Systems developed by students of the *Web Development & Semantic Web* course of our Postgraduate Program in Computer Science, all of which aim to publish Linked Data. We evaluated our proposal by creating goal and risk models for these systems and searching for vocabularies based on these models. Artifacts are available in a public source code repository: <https://github.com/nemo-ufes/FrameWeb-GRALD>.

During evaluation, we particularly focused on three research questions about GRALD: **RQ1**: can it be applied to different systems and domains? **RQ2**: can it be applied to identify risks and new related GORE elements? **RQ3**: can it aid in the identification of vocabularies?

We applied GRALD to three different systems: TravelNM (storefront for a travel agency), Transparency Portal (display government data for citizens) and Semed (information system for a medical practice). By applying GRALD on these existing systems, we were able to identify their goals, tasks, resources and actors, and build their goal models. Moreover, we were able to identify risks related to linked data and other risks, producing their risk models. These new evaluation efforts complement that of our running example, C2D, already discussed in previous chapters.

A side effect of conducting this evaluation of GRALD was the inclusion in the catalog of goals and risks for the development of Linked Data Systems (cf. Section 3.4) of goals, risks and other modeling elements that were used in these projects and might be useful in other contexts as well. Unfortunately, the catalog was being built in parallel with these evaluation efforts and, thus, has not yet been evaluated itself.

With respect to the task of choosing new vocabularies, in future work we plan to make a more in-depth analysis of the task of finding vocabularies with more equivalent properties. The execution of this task in the evaluation was only to demonstrate the applicability of the method regarding this task. An ad-hoc analysis was performed to fill out the checklist.

4.1 Travel NM

In the case of TravelNM, whose goal model is depicted in Figure 32, publication of Linked Data about cities and tour packages being offered is represented by the goal *Data published in linked data*. There is an *Agent* that represents websites that provide unified

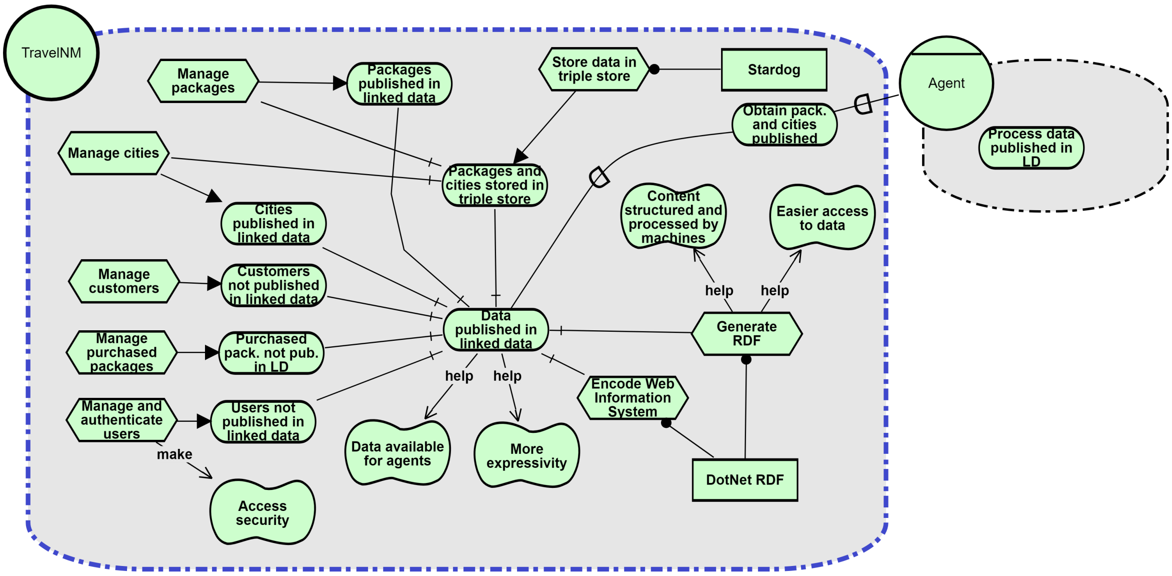

Figure 32 – iStar goal model for TravelNM.

search engines for traveling services, such as Trivago or Expedia, with a goal *Process data published*, processing the Linked Data published by *TravelNM*. Different from the goal model of C2D, TravelNM is responsible for the tasks *Generate RDF* and *Encode Web Information System* performed by the resource *DotNet RDF*.[1]

The main goal is divided in subgoals and the tasks performed are *Manage packages*, *Manage cities*, *Manage customers*, *Manage purchased packages*, *Manage and authenticate users*, *Store data in triple store*, *Encode Web Information System* and *Generate RDF*. The data is stored in a triplestore, represented by the goal *Packages and cities stored in triple store* with the task *Store data in triple store*, using the resource *Stardog*,[2] which was chosen for this project.

About the *qualities* of the system, the main goal *Data Published in Linked Data* helps TravelNM to provide *Data available for agents* such as the ones mentioned above, and have *More expressivity* because the data is published in RDF format. The task *Generate RDF* helps the qualities *Content structured and processed by machines* and *Easier access to data*.

Figure 20, which represented the *Domain Expert* actor for the C2D system, can be reused for the TravelNM, as well as the actor *Ontology Engineer*, in Figure 21. However, in the case of TravelNM, this actor executes the task *Choose Vocabulary*, not showed here. Also, in this case the actor *Programmer* is not modeled because the tasks *Generate RDF* and *Encode Web Information System* are done by the actor TravelNM system. Figure 33 shows the Strategic Dependency Diagram for the TravelNM example.

1 <http://www.dotnetrdf.org/>
2 <https://www.stardog.com/>

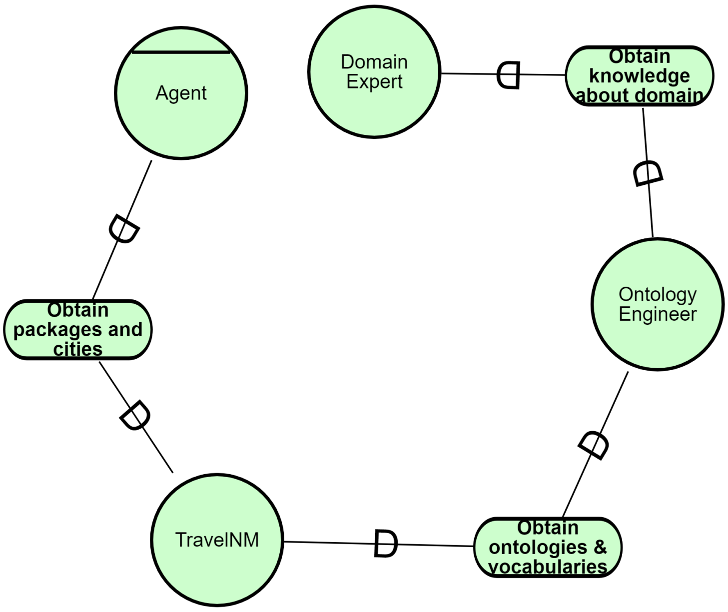

Figure 33 – iStar TravelNM goal model SD (Strategic Dependency) view.

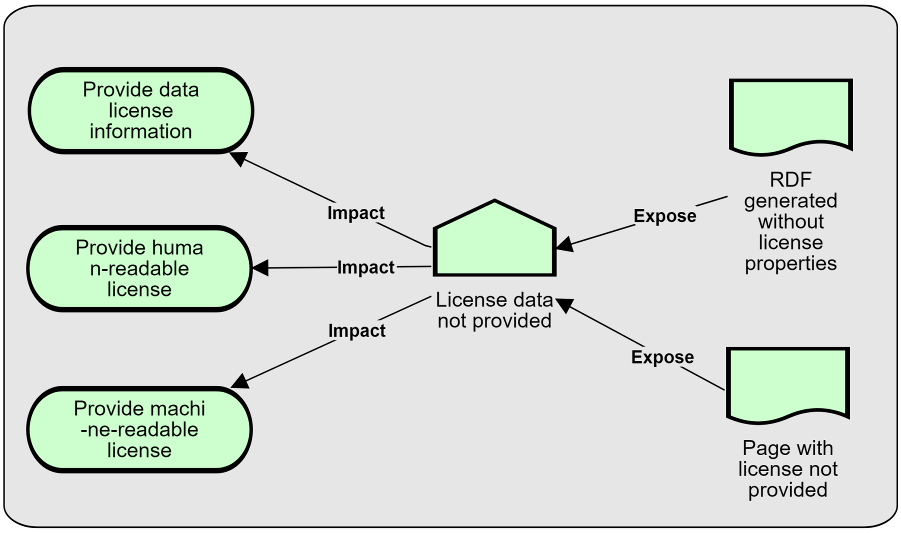

Figure 34 – RiskML risk model for TravelNM, related to RDF License

In the TravelNM system, new risks about license and SPARQL Endpoint provided will be taken into account. According to the W3C (2017), licenses provided by a data publisher may establish restrictions on the sharing of data and may be specified via meta-data or a link to a certain page. Some vocabularies such as **dct:license**, **cc:license**, **schema:license** and **xhtml:license** can be used. Based on this, Figure 34 was elaborated, in this case the license refers both to RDF documents and datasets.

In this RiskML model, situations *RDF generated without license properties* and *Page with license not provided* expose the risk event *License data not provided*, which impact the goals *Provide data license information*, *Provide machine-readable license* and *Provide human-readable license*.

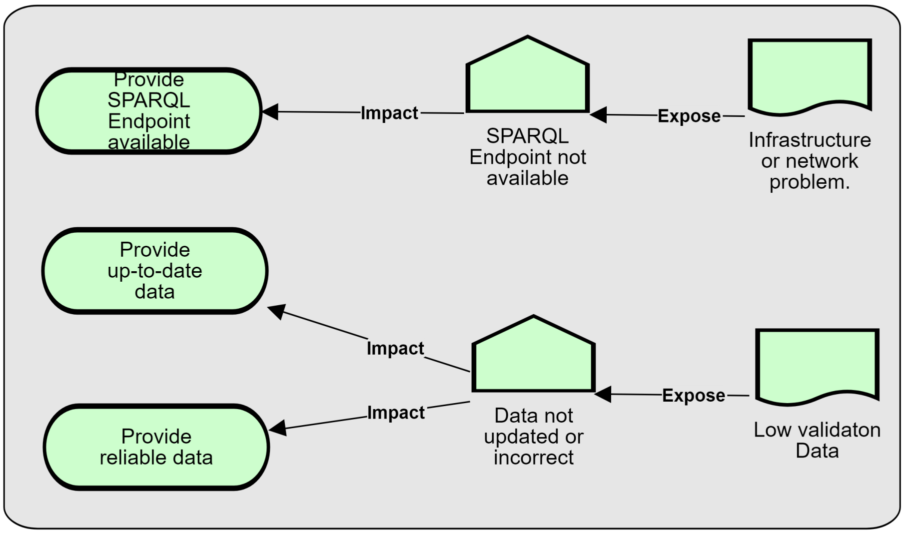

Figure 35 – RiskML risk model for TravelNM, related to SPARQL Endpoint provided

Figure 35 shows the risk model related to SPARQL Endpoint provided. The risk event *SPARQL Endpoint not available* exposed by the situation *Infrastructure or network problem* impacts the goal *Provide SPARQL Endpoint available*. The goals *Provide up-to-date data* and *Provide reliable data* are impacted by the risk event *Data not updated or incorrect* exposed by the situation *Low validation of data*.

Figure 36 presents the integrated goal model, related to the issues of dataset license and SPARQL endpoint availability. The goal *Provide data license information* is added in the model, as a task, which allows you to satisfy the goals *Machine-readable license provided* and *Human-readable license provided*, both for RDF and SPARQL endpoint data licenses. The risk event *License data not provided* impacts these elements.

Also, in the TravelNM system model, the goal *Provide SPARQL Endpoint* is added in the model, as a task, which allows you to reach the goals *Up-to-date data provided* and *Reliable data provided*, impacted by the risk events *Data not updated or incorrect* and *SPARQL Endpoint not available*.

About the vocabularies searched, for the class *City*, for instance, we found vocabularies such as `schema:City` and `dbo:City`, which were not originally found by the students when their assignment was produced. These vocabularies have already been cited in Table 4.

4.2 Transparency Portal

The Transparency Portal system addresses the issue of open data in the public sector. In Figure 38, the actor *Expenses Manager* is a system which has the main goal *Government expenses published in LD*, in order to *Keep transparency* and *Growth of social control*. For the data publication, we use the FrameWeb-LD method, therefore the actors

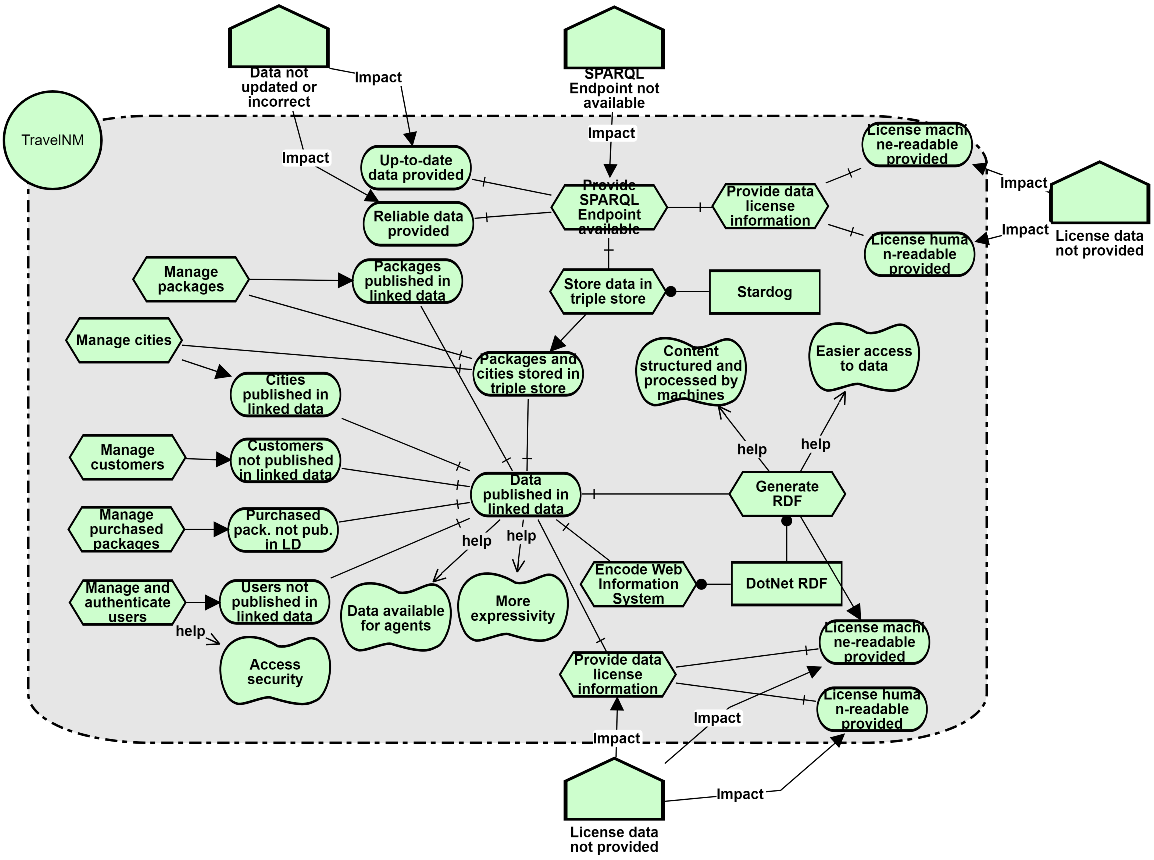

Figure 36 – Integrated Goal model for TravelNM, related to SPARQL Endpoint provided and RDF and Dataset License, connected to RiskML risk events.

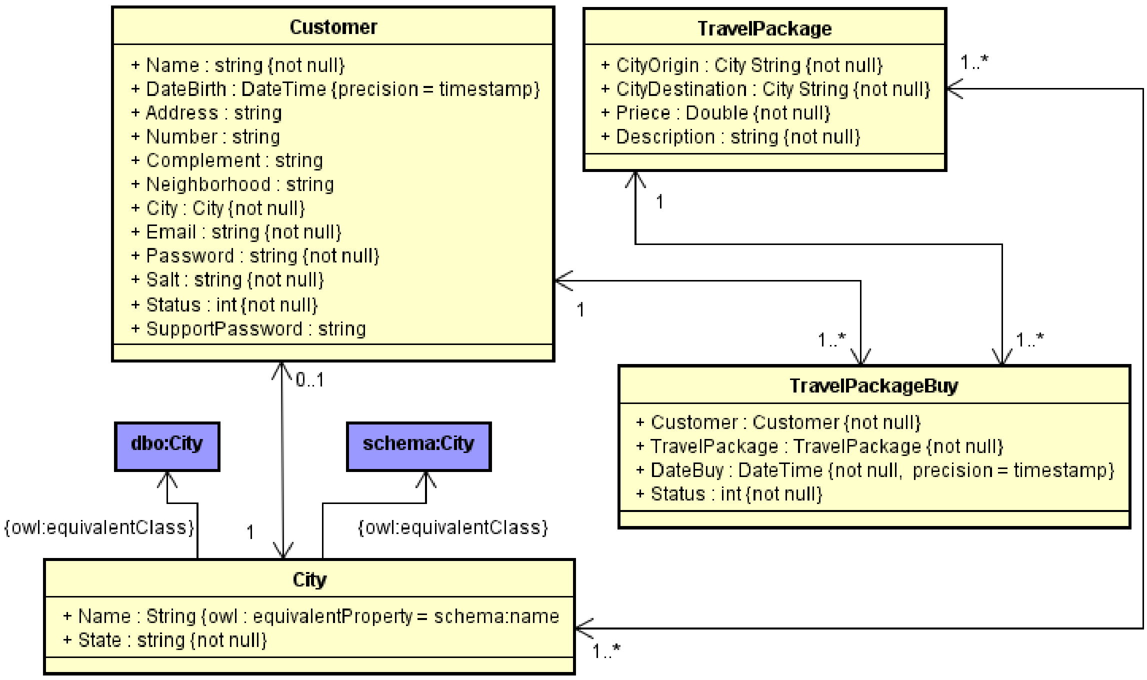

Figure 37 – The FrameWeb-LD Entity Model for Travel with newly added vocabularies during the GRALD process.

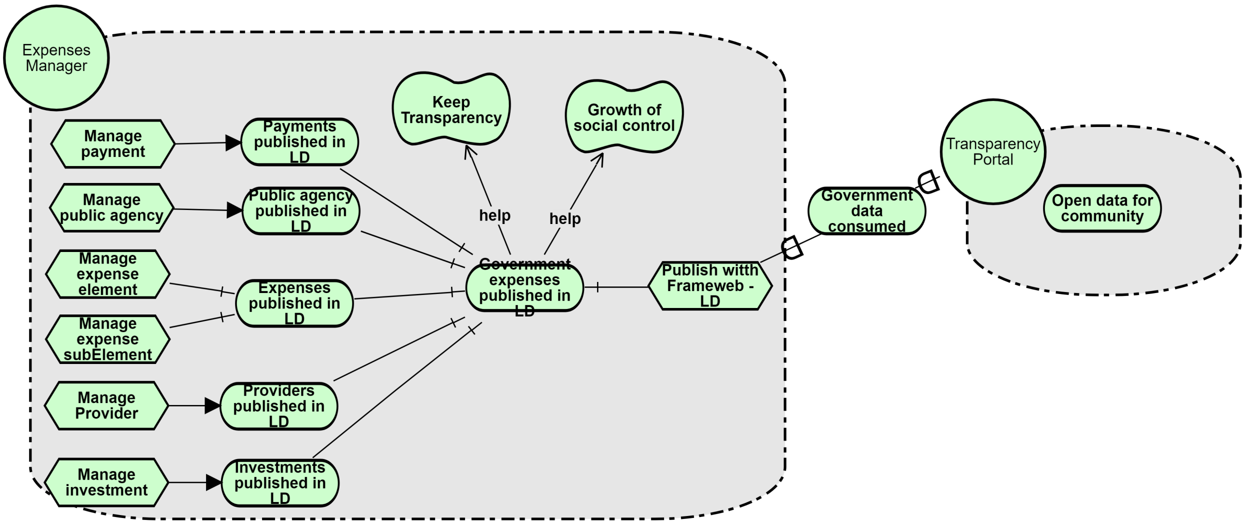

Figure 38 – iStar goal model for Transparency Portal.

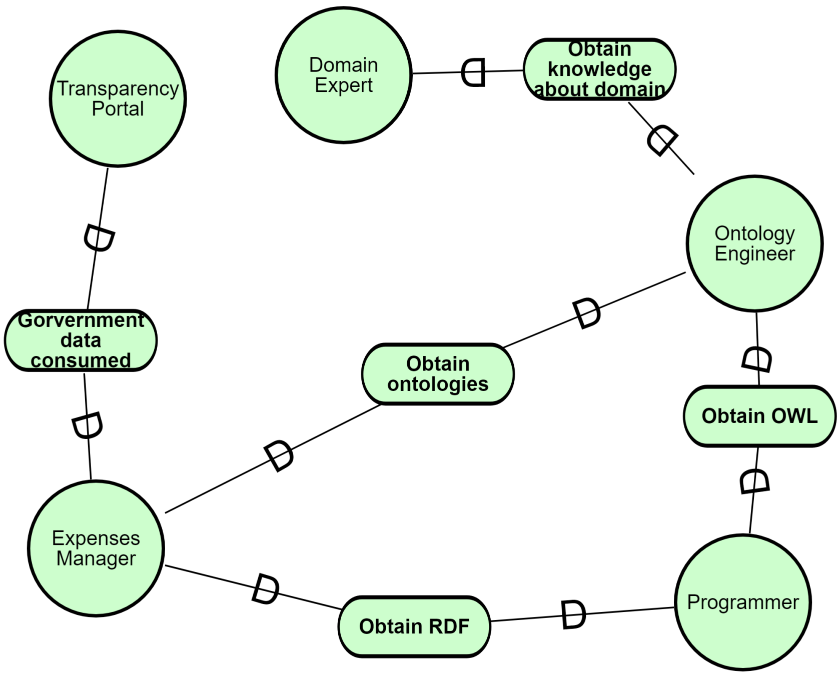

Figure 39 – iStar Transparency Portal goal model SD (Strategic Dependency) view.

Domain Expert (Figure 20), *Ontology Engineer* (Figure 21) and *Programmer* (Figure 22) also apply here. The Strategic Dependency Diagram is similar to that of the C2D system (given that universities are also part of the public sector and share common transparency goals), shown in Figure 39.

About the risks identification and models in the Transparency Portal, for illustrative purposes, risks related to data provenance will be taken into account. According to the W3C (2017), the challenge is to publish data and provide details on its origin. Through the provenance of the data, consumers can rely on the integrity and credibility of the data being shared. Based on (W3C, 2017), the Risk Model of Figure 40 was produced.

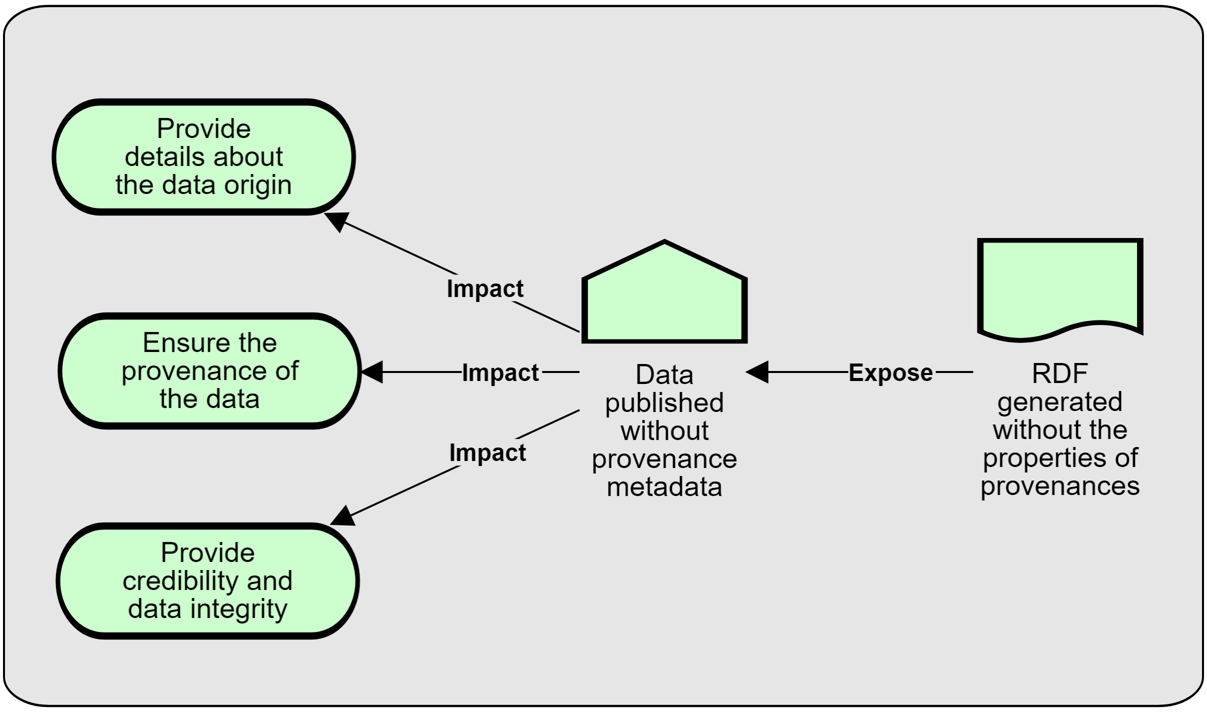

Figure 40 – RiskML risk model for Tranparency Portal, related to Data Provenance

In this model, the goals related to data provenance are *Provide details about the data origin*, *Ensure the provenance of the data* and *Provide credibility and data integrity*. The situation *RDF generated without the properties of provenances* expose the risk event *Data published without provenance metadata*. According to the W3C (2017), properties such as *dct:creator*, *dct:publisher* and *dct:issued*, in the *Data Catalog Vocabulary* (DCAT)[3] can be used to provide information about the data origin.

Figure 41 shows the integrated goal-risk model for the Transparency Portal. In the figure, the risk event *Data published without provenance metadata* impacts the new goals *Provide details about the data origin*, *Ensure the provenance of the data* and *Provide credibility and data integrity*, because an RDF generated without the properties of provenance is not in accordance with the best practices, and, in this case, machines will not be able to automatically process information of provenance (W3C, 2017).

Figure 42 shows the FrameWeb-LD Entity Model for Transparency Portal. For the class *Payment* is suggested the vocabulary *payment:Payment*[4] (Equivalent Class); for the class *PublicAgency* the vocabularies *org:OrganizationalUnit*,[5] *bio:Organization*[6] and *schema:Organization*[7] (Sub Class of) and, finally, for the class *Provider* the vocabu-

3 <https://www.w3.org/TR/vocab-dcat/>
4 <https://data.gov.uk/resources/payments/reference#ref_payment_Payment>
5 <https://www.w3.org/TR/vocab-org/#org:OrganizationalUnit>
6 <http://vocab.org/bio/>
7 <https://schema.org/Organization>

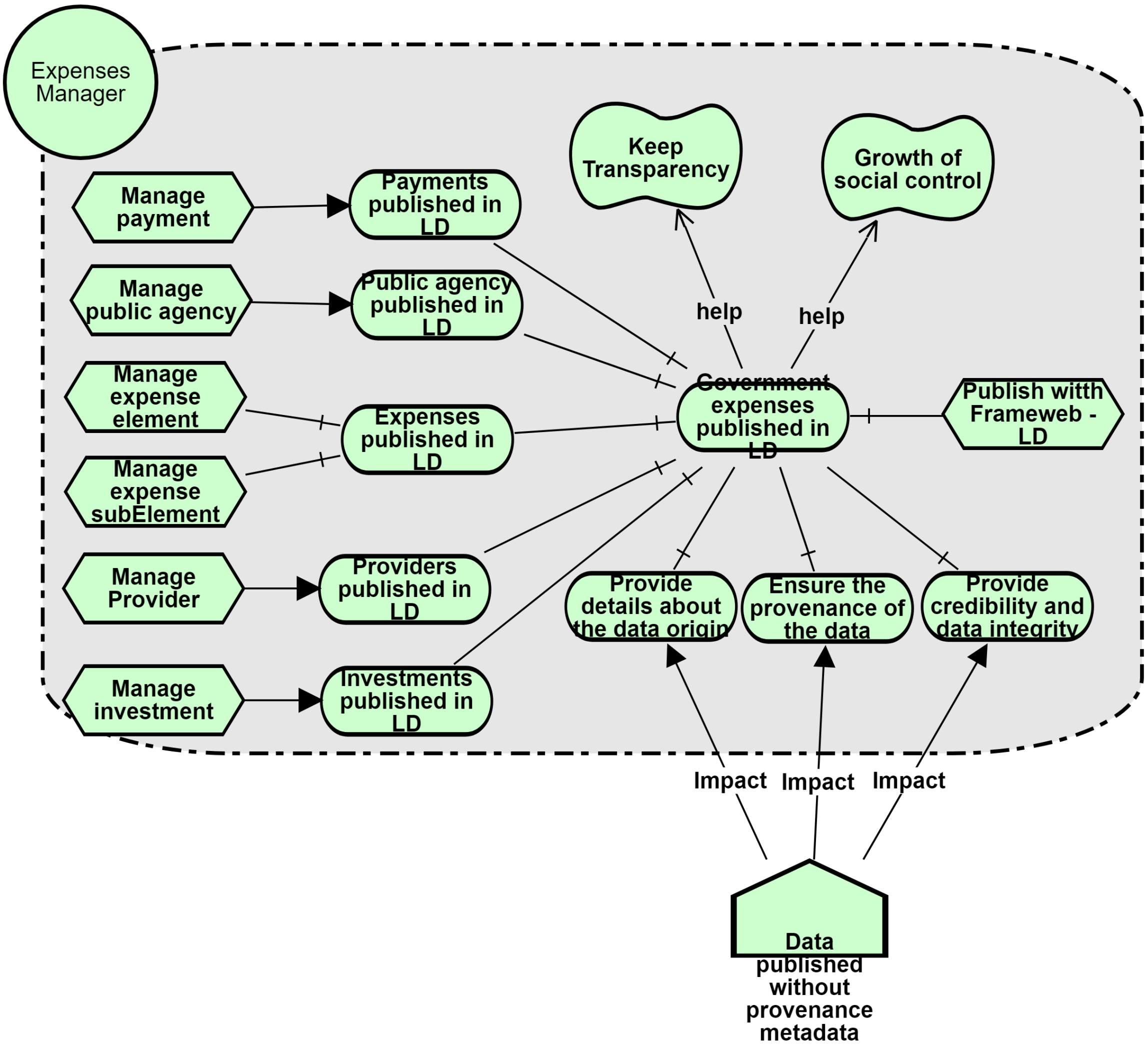

Figure 41 – Integrated Goal model for Semed, related to data provenance, connected to RiskML risk events.

laries *org:FormalOrganization*[8] and *dbo:Person*.[9] Table 5 presents the checklist for these vocabularies.

4.3 SeMed

SeMed deals with the Medical domain, as can be seen in its goal model, shown in Figure 43. The tasks *Search Drugs in Linked Data* and *Search Diseases in Linked Data* use the dataset *Bio2RDF*, and the task *Search Cities in Linked Data* use a dataset *DBPedia - OWL*. The main goal *Medical Data Published in Linked Data* helps *Open medical data for communities* and *Open medical data for patients*, and is divided in four subgoals *Drugs Published in Linked Data*, *Diseases Published in Linked Data*, *Doctors Published in Linked Data* and *Patients not published in Linked Data*, and the tasks performed are *Manage*

8 <https://www.w3.org/ns/org#FormalOrganization>
9 <http://dbpedia.org/ontology/Person>

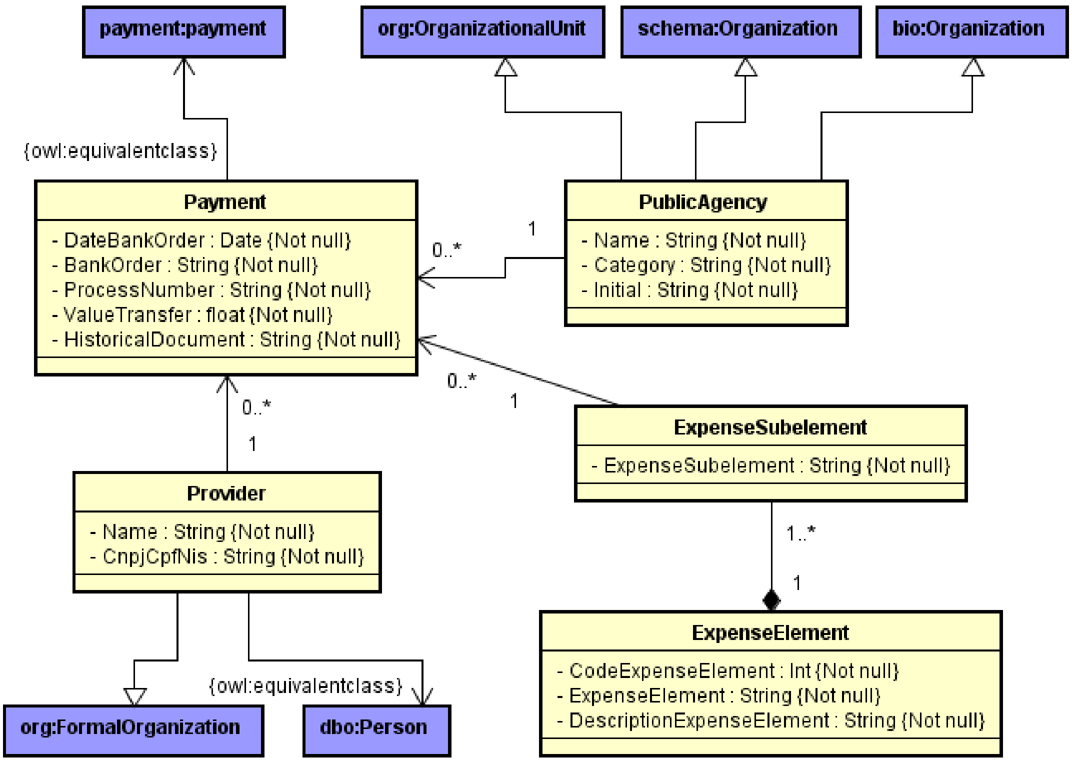

Figure 42 – The FrameWeb-LD Entity Model for Transparency Portal with newly added vocabularies during the GRALD process.

Table 5 – Vocabulary checklist for Transparency Portal.

#	Attributes	Schema	Bio	Org	Dbo	Payment
1	Published by a trusted group or organization	✓	✓	✓	✓	✓
2	Have permanent URIs	✓	✓	✓	✓	✓
3	Version control policy	✓	✓	✓	✓	✓
4	Documented vocabularies	✓	✓	✓	✓	✓
5	Self descriptive vocabularies	✓	✓	✓	✓	✓
6	Described in more than one language	×	×	✓	✓	×
7	Used by other data sets	✓	✓	✓	✓	±
8	Available for access for a long or infinite time	✓	✓	✓	✓	✓

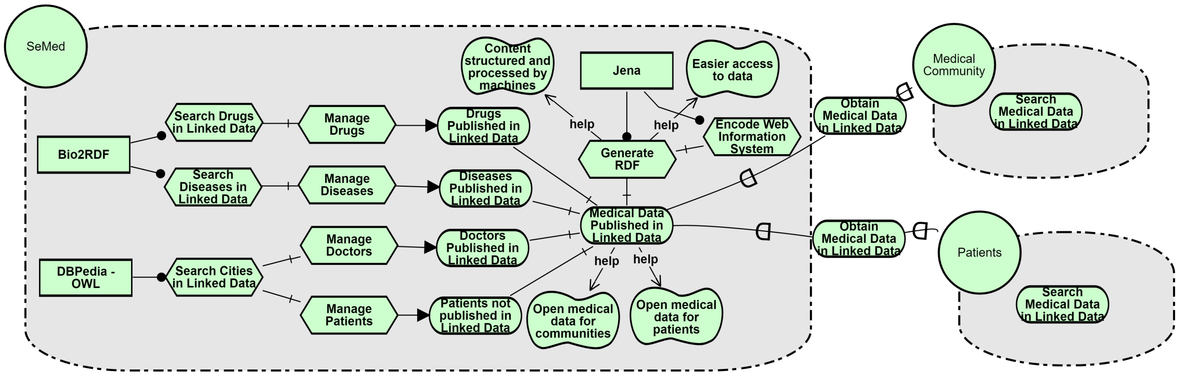

Figure 43 – iStar goal model for Semed.

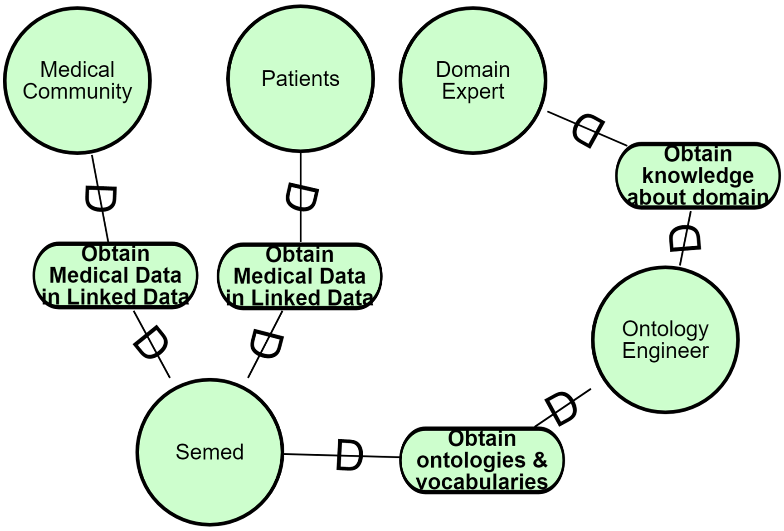

Figure 44 – iStar Semed goal model SD (Strategic Dependency) view.

Drugs, *Manage Diseases*, *Manage Doctors*, *Manage Patients*, *Generate RDF* and *Encode Web Information System*. The resource *Jena* is used to write the RDF file and Encode Web Information System.

The actors *Medical Community* and *Patients* have the goal *Search Medical Data in Linked Data*. Figure 44 shows the Strategic Dependency Diagram for this system.

Regarding risks identification and modeling, in the Semed system risks related to the choice of dataset will be taken into account, as well as the risks related to choice and adoption of vocabularies and data publication, not showed here. Figure 45, presents the risk model for the Semed system. In the Semed goal model, data related to Drugs, Diseases and Cities are searched in linked data datasets.

New goals added are: *Data obtained from specific domain* and *Registration optimized data*, impacted by the risk event *SPARQL Endpoint not available* exposed by the situation *Infrastructure or network problem*. Other goals also added are: *Reliable obtained data* and

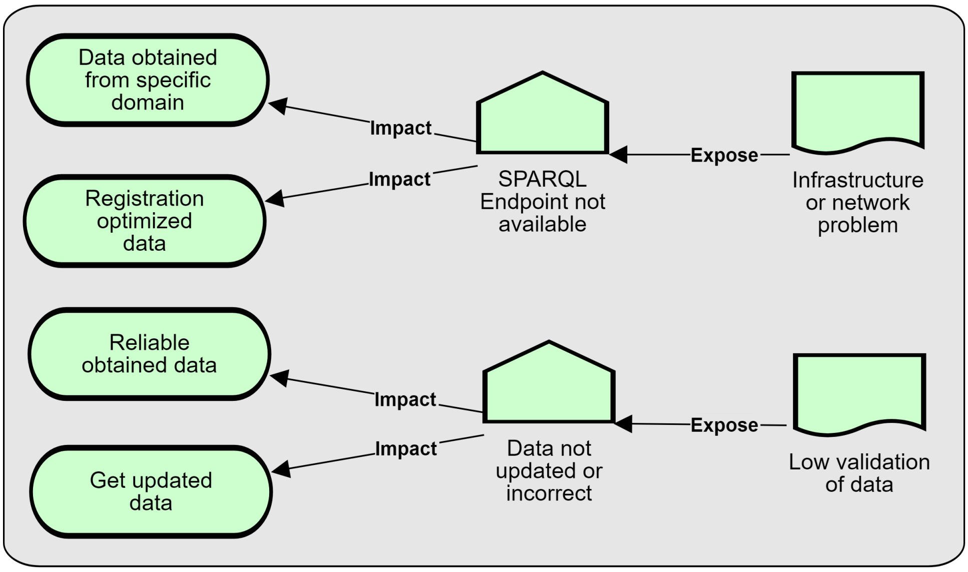

Figure 45 – RiskML risk model for Semed, related to search data in Linked Data datasets

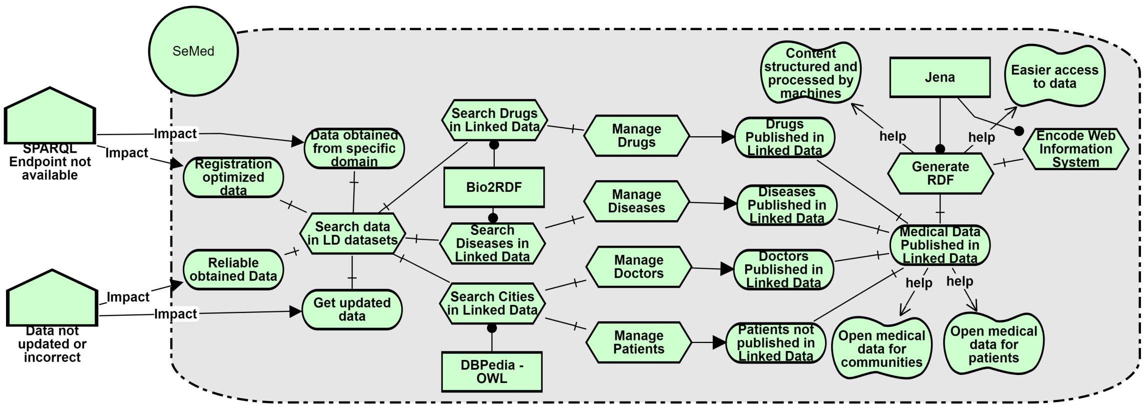

Figure 46 – Integrated Goal model for Semed, related to search in Linked Data datasets, connected to RiskML risk events.

Get updated data impacted by the risk event *Data not updated or incorrect* exposed by the situation *Low validation of data*.

Figure 46 shows the integrated goal model for the Semed system related to search in Linked Data datasets. In this figure, newly added elements include the task *Search data in LD datasets* and goals *Data obtained from specific domain*, *Registration optimized data*, *Reliable obtained Data* and *Get updated data*, impacted by risk events *SPARQL Endpoint not available* and *Data not updated or incorrect*, as explained before.

The risk event *SPARQL Endpoint not available* impacts the goals *Data obtained from specific domain*, *Registration optimized data* because it will disrupt the registration of Drugs, Diseases, Doctors and Patients. The risk event *Data not updated or incorrect* impacts

10 <http://schema.org/Person>
11 <http://xmlns.com/foaf/spec/#term_Person>
12 <http://dbpedia.org/page/Physician>

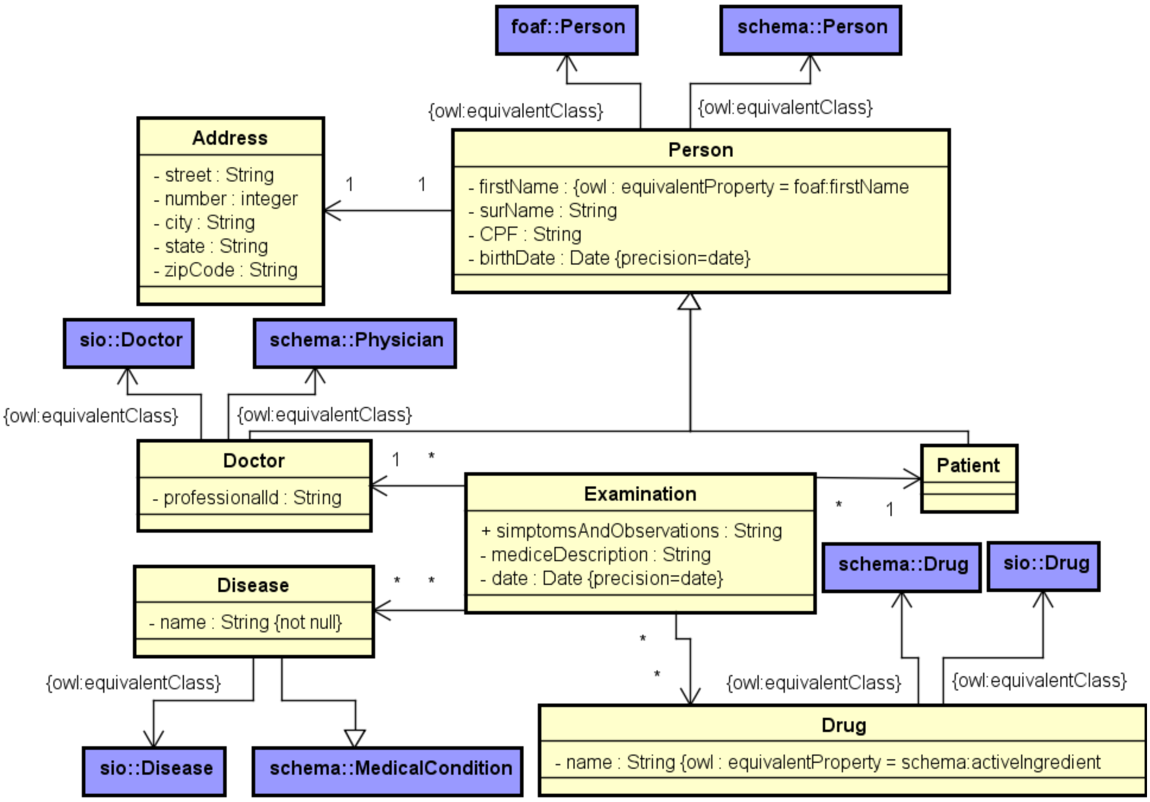

Figure 47 – The FrameWeb-LD Entity Model for Semed with newly added vocabularies during the GRALD process.

the *Reliable obtained Data* and *Get updated data* because the veracity and timeliness of the data are very important for searching data in Linked Data datasets.

Regarding vocabulary search, for the class *Person*, for instance, we found vocabularies *schema:Person*[10] and *foaf:Person*;[11] for the class *Doctor*, *dbo:Physician*[12] and *sio:Doctor*;[13] for the class *Disease*, *schema:MedicalCondition*[14] and *sio:Disease*;[15] and, finally, for the class *Drug*, *schema:Drug*[16] and *sio:Drug*.[17] Figure 47 shows the FrameWeb-LD Entity Model for Semed. Table 6 presents the checklist for these vocabularies.

4.4 Conclusion of this Chapter

Finally, we analyze the proposed research questions:

RQ1: Can GRALD be applied to different systems and domains? The systems in which GRALD was successfully applied during this evaluation involved different domains, such as education, geographical, government, etc., which indicates a positive

13 <http://semanticscience.org/resource/SIO_000394>
14 <http://health-lifesci.0.3-3b.schemaorgae.appspot.com/MedicalCondition>
15 <http://semanticscience.org/resource/SIO_010299>
16 <http://health-lifesci.0.3-3b.schemaorgae.appspot.com/Drug>
17 <http://semanticscience.org/resource/SIO_010038>

Table 6 – Vocabulary checklist for Semed.

#	Attributes	Schema	Sio	Foaf
1	Published by a trusted group or organization	✓	✓	✓
2	Have permanent URIs	✓	✓	✓
3	Version control policy	✓	✓	✓
4	Documented vocabularies	✓	✓	✓
5	Self descriptive vocabularies	✓	✓	✓
6	Described in more than one language	×	×	×
7	Used by other data sets	✓	✓	✓
8	Available for access for a long or infinite time	✓	✓	✓

answer to this RQ.

RQ2: Can GRALD be applied to identify risks and new related GORE elements? Applying GRALD to the aforementioned systems, although very simple and small, we were able to elicit and model risk elements, then augment the goal model with new elements (goals) related to these risks. Further risks could be found with the use of risk identification techniques that are out of the scope of this work.

RQ3: Can GRALD aid in the identification of vocabularies? GRALD activities *Elicit Requirements*, *Develop Domain Model* and *Develop Goal Model* allowed us to model the classes of the system and clearly specify those that will have the published objects in Linked Data.

Limitations: although the evaluation answers positively our research questions, it suffers from the following limitations:

- The evaluation was conducted by the author himself and one undergraduate student, under his supervision. Better results are achieved by having third parties (students, practitioners) apply the method;

- The catalog of goals and risks was developed in parallel with the evaluation efforts and, thus, was not used during evaluation. It should be assessed if the catalog actually helps practitioners in the development of Linked Data Systems;

- The models were produced based on projects developed by graduate students of the *Web Development & The Semantic Web* course by reverse engineering the systems that had already been developed. Thus, it is not possible to properly assess the usefulness of the method for practitioners (a research question that is still open for us);

- In hindsight, we believe that we could have benefited from a more careful planning of the methodological aspects of the evaluation. Since RISCOSS and FrameWeb-LD already had their own applicability evaluations conducted, more focus should have been given on the utility of GRALD, which combines the other two approaches, in practice. This, however, is left for future work.

In this chapter, the evaluation of the proposal was done and, despite the above limitations, it shows that the approach is promising. The activities of goal modeling, identification and modeling of risks, choice of vocabularies and creation of the entity model based on the FrameWeb-LD approach were accomplished.

5 Conclusions

In this work, we presented GRALD, *Goal and Risk Analysis for Linked Data*, an approach based on RISCOSS, which applies Goal-Oriented Requirements Engineering (GORE), using iStar, for the development of Web-based Information Systems that publish Linked Data, using FrameWeb-LD, integrating goal models with risk models in order to perform risk analysis.

GRALD assists developers, designers and researchers with regards to the state of practice because, in GRALD, GORE is applied in order to help developers to analyze their system objectives, as well as the goals and actors related to the implementation of Linked Data, mapping the necessary resources and tasks to accomplish it. Moreover, performing risk analysis helps to analyze the impact of the occurrence of risk events on system/business goals, as well as to carry out the prevention/mitigation of these risks. Finally, GRALD assists developers in the choice of vocabularies based on the tasks performed in the phases of early and late requirements, having the search of such vocabularies accomplished using Linked Data search engines following guidelines from a checklist. From the creation of the models, a catalog of goals and risks for the publication of data in Linked Data was elaborated, with the objective of serving as a basis for the creation of new models, as well as documentation for models already created.

GRALD can be used in several application domains, but as a challenge related to its use in practice are:

- The knowledge about the concepts approached, related to GORE, RiskML and FrameWeb-LD;
- Knowledge related to the application domain, as well as abstraction capability needed to identify risks, goals, resources, tasks, etc.
- Learning how to use tools for modeling.

The adoption of vocabularies for the publication of interconnected data is a very important task, and based on W3C good practices, this work suggested a way to help developers choose the best vocabularies for their Linked Data systems. We consider, therefore, that the objectives of this work have been achieved. The goal-oriented language iStar (cf. Section 2.3) and risk modeling language RiskML (cf. Section 2.5) were used; a tool to aid in the design of goal and risk models was provided (cf. Section 3.2.5); a checklist to aid developers in the choice of Linked Data vocabularies was proposed (cf. Section 3.3.1); a catalog of risk and goals for the implementation of Linked Data Systems was offered (cf. Section 3.4), finally the approach was evaluated (cf. Chapter 4).

The **strengths** of this work are:

- The use of iStar to create goal models, allowing the identification of actors, goals, qualities, tasks, resources and dependencies in the publication of interconnected data;
- Identification of risks and their modeling with RiskML, creation of the integrated model (Goals and Risks) supporting risk analysis, related to the impact of a risk event on a goal;
- Through the identification and the modeling of risks, developers can assess and mitigate failures in the project;
- Creating risk models in categories, for example: adoption of vocabularies, data publication, data provenance and SPARQL endpoint provided, thus allowing the reuse of the models;
- Tools are provided to aid developers in the creation of all of the models prescribed by the approach;
- Suggests a catalog of goals and risks related to the Linked Data system, as a way to serve as a knowledge base for the creation of the models and to describe the elements of them;
- The use of a checklist for choosing vocabularies.

In this dissertation, the following **weaknesses** were considered:

- Few risks related to the Linked Data system were identified, precisely because of the lack of an existing database or catalog of risks;
- A tool for choosing vocabularies based on checklist, during the creation of entity models was not developed;
- The scalability of the models was not evaluated, because it is an obstacle to the goals, risks and integrated models;
- The approach has not been fully evaluated, especially with respect to its usefulness and application by practitioners.

Based on the above limitations, the following suggestions for **future work** are presented:

- Identify more risks related to the Linked Data system and create a repository of risk indicators;
- Develop a tool integrated with Linked Data search engines (e.g., LOV) to assist developers in the task of choosing vocabulary, during the creation of entity models;
- Evaluate the usefulness of the approach by having practitioners apply it in real scenarios;
- Carry out the evaluation with other systems, of greater size, to evaluate the scalability of the models.

Bibliography

AL, A.; CHOWDHURY, M.; AREFEEN, S. Software risk management: Importance and practices. 2011. Cited on page 34.

ANALYTICS, I. C. for D. *The Linked Open Data Cloud, <https://lod-cloud.net/> (last access: Jan 12th, 2019).* 2019. Cited on page 18.

ANTON, A. I. Goal-based requirements analysis. In: *Proceedings of the 2Nd International Conference on Requirements Engineering (ICRE '96).* : IEEE Computer Society, 1996. (ICRE '96), p. 136–. Cited on page 29.

ASNAR, Y.; GIORGINI, P.; MYLOPOULOS, J. Goal-driven risk assessment in requirements engineering. *Requir. Eng.*, Springer-Verlag New York, Inc., v. 16, n. 2, p. 101–116, jun. 2011. Cited 2 times in pages 18 and 45.

BANNERMAN, P. L. Risk and risk management in software projects: A reassessment. *Journal of Systems and Software*, v. 81, n. 12, p. 2118 – 2133, 2008. Cited 2 times in pages 33 and 34.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. *Scientific American*, v. 284, n. 5, p. 34–43, 2001. Cited 3 times in pages 17, 23, and 26.

BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked data-the story so far. *Int. J. Semant. Web Inf. Syst.*, v. 5, n. 3, p. 1–22, 2009. Cited 5 times in pages 9, 17, 24, 25, and 27.

BOEHM, B. W. Software risk management: principles and practices. *IEEE Software*, v. 8, n. 1, p. 32–41, Jan 1991. Cited 2 times in pages 33 and 34.

BRESCIANI, P. et al. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, v. 8, n. 3, p. 203–236, May 2004. Cited on page 29.

BRUWER, R.; RUDMAN, R. Web 3.0: governance, risks and safeguards. *Journal of Applied Business Research*, The Clute Institute, v. 31, n. 3, p. 1037, 2015. Cited on page 47.

CELINO, D. R. et al. A Framework-based Approach for the Integration of Web-based Information Systems on the Semantic Web. In: *Proc. of the 22nd Brazilian Symposium on Multimedia and the Web.* : ACM, 2016. p. 231–238. Cited 14 times in pages 9, 20, 41, 42, 43, 44, 45, 47, 52, 54, 60, 62, 67, and 71.

COSTAL, D. et al. Aligning Business Goals and Risks in OSS Adoption. In: *Proceedings of the 34th International Conference on Conceptual Modeling.* : Springer, 2015. p. 35–49. Cited 13 times in pages 9, 11, 18, 20, 34, 36, 37, 38, 39, 40, 41, 45, and 57.

DALPIAZ, F.; FRANCH, X.; HORKOFF, J. iStar 2.0 Language Guide. Jun 2016. Cited 4 times in pages 9, 31, 32, and 33.

FALBO, R. A. SABiO: Systematic Approach for Building Ontologies. In: GUIZZARDI, G.

et al. (Ed.). *Proc. of the Proceedings of the 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering.* : CEUR, 2014. Cited 8 times in pages 41, 47, 49, 51, 52, 62, 66, and 67.

FREITAS, M. L. de et al. Goal and Risk Analysis in the Development of Information Systems for the Web of Data. In: *Proc. of the 21st Ibero-American Conference on Software Engineering, Requirements Engineering track.* Bogota, Colombia: , 2018. Cited 2 times in pages 20 and 45.

GUARINO, N. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy.* 1st. ed. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1998. Cited on page 25.

GUARINO, N.; OBERLE, D.; STAAB, S. What is an ontology? p. 1–17, 05 2009. Cited on page 25.

GUERSON, J. et al. Ontouml lightweight editor: A model-based environment to build, evaluate and implement reference ontologies. In: . : IEEE Computer Society, 2015. p. 144–147. Cited on page 41.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models.* Tese (PhD Thesis) — University of Twente, The Netherlands, 2005. Cited 5 times in pages 26, 38, 42, 49, and 50.

GUIZZARDI, G. On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In: . : IOS Press, 2007. p. 18–39. Cited on page 26.

HEATH, T.; BIZER, C. *Linked Data: Evolving the Web into a Global Data Space.* : Morgan & Claypool Publishers, 2011. (Synthesis Lectures on the Semantic Web: Theory and Technology). Cited on page 17.

HORROCKS, I. Semantic web: The story so far. In: *Proceedings of the 2007 International Cross-disciplinary Conference on Web Accessibility (W4A).* : ACM, 2007. (W4A '07), p. 120–125. Cited on page 23.

HORROCKS, I. Ontologies and the semantic web. *Communications of the ACM*, Association for Computing Machinery (ACM), v. 51, n. 12, p. 58, dec 2008. Cited on page 26.

HYLAND, B.; ATEMEZING, G.; VILLAZóN-TERRAZAS, B. *Best Practices for Publishing Linked Data, <https://www.w3.org/TR/ld-bp/> (last access: May 26th, 2017).* 2014. Cited 6 times in pages 47, 60, 62, 63, 67, and 68.

JURETA, I. J. et al. Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling. In: . 2010. p. 115–124. Cited on page 29.

LAMSWEERDE, A. V. Goal-oriented requirements engineering: A guided tour. In: *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering.* : IEEE Computer Society, 2001. (RE '01), p. 249–. Cited 2 times in pages 18 and 29.

LAMSWEERDE, A. van; LETIER, E. Handling obstacles in goal-oriented requirements engineering. *IEEE Trans. Softw. Eng.*, IEEE Press, v. 26, n. 10, p. 978–1005, out. 2000. Cited 3 times in pages 18, 29, and 45.

LAPOUCHNIAN, A. Goal-oriented requirements engineering: An overview of the current research. 01 2005. Cited 2 times in pages 18 and 28.

LEHTIRANTA, L. Risk perceptions and approaches in multi-organizations: A research review 2000–2012. *International Journal of Project Management*, v. 32, n. 4, p. 640 – 653, 2014. Cited on page 33.

LÓPEZ, L. *Welcome to the RISCOSS Risk Modeling repository, <https://github.com/RISCOSS/riscoss-risk-modeling/wiki> (last access: May 26th, 2017)*. 2015. Cited 4 times in pages 9, 35, 36, and 48.

LÓPEZ, L.; SIENA, A. *How to create a Risk Model, <https://github.com/RISCOSS/riscoss-risk-modeling/wiki/How-to-create-a-Risk-Model> (last access: May 25th, 2017)*. 2015. Cited 2 times in pages 9 and 35.

MARTINS, B. F.; SOUZA, V. E. S. A Model-Driven Approach for the Design of Web Information Systems based on Frameworks. em Proc. of the 21st Brazilian Symposium on Multimedia and the Web, 2015. Cited 2 times in pages 41 and 42.

MYLOPOULOS, J.; CHUNG, L.; NIXON, B. Representing and using nonfunctional requirements: a process-oriented approach. *IEEE Transactions on Software Engineering*, v. 18, n. 6, p. 483–497, Jun 1992. Cited 2 times in pages 18 and 29.

PIMENTEL, J.; CASTRO, J. piStar Tool – A Pluggable Online Tool for Goal Modeling. In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*. 2018. p. 498–499. Cited on page 59.

POHOREC, S.; ZORMAN, M.; KOKOL, P. Analysis of approaches to structured data on the web. *Comput. Stand. Interfaces*, Elsevier B.V., v. 36, n. 1, p. 256–262, 2013. Cited 2 times in pages 17 and 24.

SILVA, A. A. *C2D - Módulo de Credenciamento e Classificação de Docentes do Sistema Marvin*. Monografia (Undergraduate Project) — Federal University of Espírito Santo, 2017. Cited on page 73.

TEKLI, J. et al. Semantic to intelligent web era: Building blocks, applications, and current trends. In: *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*. : ACM, 2013. (MEDES '13), p. 159–168. Cited 5 times in pages 9, 23, 24, 25, and 26.

W3C. *OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (Second Edition), <https://www.w3.org/TR/owl2-syntax/> (last access: May 07th, 2018)*. 2012. Cited on page 26.

W3C. *RDF Schema 1.1, <https://www.w3.org/TR/rdf-schema/> (last access: May 03th, 2018)*. 2014. Cited on page 25.

W3C. *Data on the Web Best Practices, <https://www.w3.org/TR/dwbp/> (last access: Jul 25th, 2018)*. 2017. Cited 6 times in pages 47, 62, 65, 75, 78, and 79.

W3C. *Data - W3C, <https://www.w3.org/standards/semanticweb/data> (last access: April 30th, 2018)*. 2018. Cited on page 27.

W3C. *Semantic Web - W3C, <https://www.w3.org/standards/semanticweb/> (last access: April 28th, 2018)*. 2018. Cited on page 24.

WESTFALL, L.; ROAD, C. Software Risk Management. *Risk Manag.*, p. 1–8, 2001. Cited on page 34.

YU, E. S. Social Modeling and i*. v. 5600, 2009. Cited 7 times in pages 9, 18, 20, 29, 30, 31, and 32.