8-17-2020 1:00 PM

Ontology-Driven Semantic Data Integration in Open Environment

Islam M. Ali, *The University of Western Ontario*

Supervisor: McIsaac, Kenneth, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering

Recommended Citation

Ali, Islam M., "Ontology-Driven Semantic Data Integration in Open Environment" (2020). *Electronic Thesis and Dissertation Repository*. 7230.
https://ir.lib.uwo.ca/etd/7230

Abstract

Collaborative intelligence in the context of information management can be defined as "A shared intelligence that results from the collaboration between various information systems". In open environments, these collaborating information systems can be heterogeneous, dynamic and loosely-coupled. Information systems in open environment can also possess a certain degree of autonomy. The integration of data residing in various heterogeneous information systems is essential in order to drive the intelligence efficiently and accurately. Because of the heterogeneous, loosely-coupled, and dynamic nature of open environment, the integration between these information systems in the *data level* is not efficient. Several approaches and models have been proposed in order to perform the task of data integration. Many of the existing approaches for data integration are designed for closed environment, tightly-coupled systems and enterprise data integration. They make explicit, or implicit, assumptions about the semantic structure of the data. Because of the heterogeneous and loosely-coupled nature of open environment, such assumptions are deemed unintuitive. Data integration approaches based on model that are extensional in nature are also inadequate for open environment. This is because they do not account for the dynamic nature of open environment. The need for an adequate model for describing data integration systems in open environment is quite evident. Intensional based modeling is found to be an adequate and natural choice for modeling in open environment. This is because it addresses the dynamic and loosely-coupled nature of open environment. In this work, an intensional model for the conceptualization is presented. This model is based on the theory of Properties Relations and Propositions (PRP). The proposed description takes the concepts, relations, and properties as primitive and as such, irreducible entities. The formal intensional account of both Ontology and Ontological Commitment are also proposed in light of the intensional model for conceptualization. An intensional model for ontology-driven mediated data integration in open environment is also proposed. The proposed model accounts for the dynamic nature of open environment and also intensionally describes the information of data sources. The interface between global and local ontologies and the formal intensional semantics of the query answering are then described.

Keywords

Collaborative Intelligence, Data Integration, Data Integration Systems, Open Environment, Conceptualization, Ontology, Ontological Commitment, Extensional Logic, Intensional Logic, Extensionalization, Intensional Model, Epistemic Logic, Intensional Epistemic Logic.

Summary for Lay Audience

In today’s world, data can be found anywhere, databases, web pages, email inboxes, and many more types of data sources. Some of these data sources are structured, i.e. they have tables and fields, like the case with databases. Other data sources are unstructured. This is the case with information that reside on a webpage or in your email inbox. This means that these data sources are heterogeneous. Another factor that affects the heterogeneity is the fact that, even the structured data sources are created by different parties. These various parties created their data sources with different needs in mind. And so, they tailored the data source to satisfy these particular needs. When it comes to generating intelligence for the purpose of driving decision making, one should attempt to take advantage of all available data sources. For example, it has been found that most of the information about customer satisfaction/frustration with a business can’t be found in an enterprise database. Rather, most of this information is on web pages, blogs, forums, or in the email inbox of a customer care representative. Nowadays also the communication on the web is very dynamic. Agents, computers, phones, servers, and other equipments can connect/disconnect from the web at anytime. This is an example for what we refer to as an open environment. In open environment agent can enter and leave the environment at anytime and the environment should still continue to function. As mentioned earlier, in order to generate intelligence, one should attempt to utilize the data from various data sources. In order to do so, the data from the various data sources need to be aligned and combined somehow. This can be referred to as data integration. In this work, we propose a model for data integration that accounts for the characteristics of what is referred to earlier as open environment.

Acknowledgments

I want to express my deep gratitude to my advisor Prof. Kenneth McIsaac for his continues and endless support for several years. Prof. McIsaac’s fruitful and constructive comments have been very helpful and added noticeable value to my work. I am also grateful to Anh Brown, Joan Aldis, Dr. Wendy Dickinson, and Jennifer Meister for helping me navigate through the university. I also offer many thanks to my friend and brother Dr. Ahmed Alassuity for his compassion and perseverance and his trust in my work. I also want to thank my loving wife Safae Malki who has been my companion and a driving force behind my success. I am also grateful to my father, Mohamed Abdelsalam who gave me all the love, care, and confidence that I needed to complete this work. And Finally, I want to thank my late mother Nazlah Abu Zaid who never stopped giving until she died. She had one wish before she died, she wanted to attend my graduation commencement. I ask God to deliver the news of my graduation to her in Paradise.

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

Intelligence is the main driver for decision support, forecast, and business process management. It does play a very crucial rule on both the commercial and scientific levels. Extracting intelligence from various heterogeneous data sources in open environment requires the integration between these data sources. Because of the heterogeneity and dynamic nature of open environment, the integration approaches need to account for the heterogeneous and dynamic nature. In this chapter, we will shed some light on the definitions of collaborative intelligence, data integration and the elements of data integration systems. We will also, briefly, discuss related work in the area of semantic data integration. Research issues and objectives will then be identified.

1.1 Collaborative Semantic Intelligence

In the context of knowledge modeling, the definition of *intelligence* is based on three principles. These principles are; *Data*, *Information*, and *Knowledge* (Makhfi 2007).

- Data is defined as, the measures and symbols of the world around us (Makhfi 2007). It is presented as external signals and picked up by various sensory instruments and organs. In order to make it clear, think about raw signals, voltages, number, distances, positions, or other physical quantities. They all represent data. For example; beeps in Morse code are considered data.

- *Information*, is produced when meanings are attached to data. In that sense, *data* becomes *information* when it becomes relevant to our decision-making process (Makhfi 2007). For example; the beeps in Morse code stand for “S-O-S”.

- *Knowledge*, is the subjective interpretation of *Information* in effort to recognize the applications and approach to act upon in the mind of perceiver. As such, *Knowledge* attaches purpose to *Information*, resulting in the potential to generate action (Makhfi 2007).

- *Intelligence* is wisdom which embodies awareness, insight, moral judgments, and principles to construct new *knowledge* and improve upon one’s existing *Knowledge*.

Example: Think about a measure, “8,848 meters” is *data*. It is not very meaningful, probably not something you can reason about. You can attach a meaning to this data by saying “The height of Mount Everest is 8,848 meters”. Now the statement “The height of Mount Everest is 8,848 meters” is *information*. It is clear that the statement informs you about the height of Mount Everest. As such, it is something you can reason about and is relevant to decision making. We can attach a purpose to the above statement by adding the rule “If the height of Mount Everest is more than 5,000 meters, then do not climb”. Now there is a purpose attached to the information that made it relevant to making a decision, to climb or not to climb. This is what we call *knowledge*. As for *intelligence*, it is reasoning, judging, and making a decision given the knowledge. In this case, and give n the knowledge above, the result of reasoning is “To not climb”. This example helps understanding the relationship and differences between *data*, *information*, *knowledge*, and *intelligence*.

The intelligence defined above is associated to one individual or agent. However, there are other types of intelligence that involve more than one agent or more than one individual. For instance; collective intelligence is defined as the ability of a group to solve more problems than its individual members (Heylighen 1999). In that sense, the organizations and teams are built on the assumption that their members can do together more than each member would do alone. Collective intelligence is also defined as, a groups of individuals acting collectively in ways that seem intelligent (Malone 2008). Another type of intelligence that characterizes distributed systems is the collaborative intelligence. Collaborative intelligence characterizes multi-agent, distributed systems where each agent is uniquely positioned with autonomy to contribute to a problem-solving network (Gill 2012). In open environments, the collaborating agents can be heterogeneous, dynamic, and possess certain degree of autonomy. In the context of information systems, the integration between various information systems is necessary in order to achieve the goal of collaborative intelligence. Some data integration techniques and frameworks the integration between various data sources is done at the data level. In open environment, however, the integration between various information systems at the data level is not efficient. This is because of the heterogeneous nature of the environment. As such, in open environment, there is no control over the data residing in data sources or the beliefs of agents. And so, a mechanism is required in order to collaboratively drive intelligence from the carious heterogeneous information systems efficiently and accurately. This can be done through the use of semantics. And if we assume that the different information systems are built for the same domain, then the integration between the various information systems, in the semantic level, is possible (Xue 2010). And in turns, driving the intelligence collaboratively in the semantic, or conceptual, level can also be attainable. The assumption here is that various information systems share the same conceptualization for some domain of interest. Each information system may have different representation for that conceptualization. However, despite the different representations, the semantics work as the common ground between these information systems. The semantics can be implicit or explicit. For example; the schemas of a relational database contains implicit semantics. The semantics extracted from a database schema, however, are not as accurate as the explicit semantics. On the other hand, ontologies do provide explicit semantics. Explicit semantics are indeed more accurate and up to date as opposed to implicit semantics.

1.2 Ontology-Based Data Integration

Data integration is a very important tool for driving collaborative intelligence. It is the process of combining data residing at various data sources and presenting the user with a collaborative view of the data. The integration of data can be physical or virtual. Physical data integration techniques tend to create a common physical data store or data repository in which data are consolidated. In this type of integration, the common data repository needs to be updated as soon as one of the data sources is updated. This is important for the integrity of the data. On the other hand, virtual data integration creates unified, logical virtualized views. When virtual data integration is used, there is no need to move the data to a common data store. Instead, the schemas or ontologies of various data sources are aligned so that a unified view of the data is possible. This can save the troubles associated with the need to keep the data repository up to date when the data sources are updated. Also, because of the dynamic nature of open environment, there is no guarantee that data sources are going to be available at all times. As such, the system needs to continue to function given whatever data sources available. Because of the dynamic nature of open environment, virtual data integration techniques are more appropriate for the task of data integration in open environment. Data integration can also be achieved at the data level or at the conceptual layer. Because of the heterogeneous nature of open environment, there is no control over the data residing at each data source. As such, data integration at the semantic level is more appropriate for open environment settings. And finally, the semantic data integration can be done in various ways. Some technique use schema matching techniques. Other techniques perform the integration at the level of ontology. Even though a data source schema contains semantics, the semantics derived from a data schema is implicit and not maintainable. On the other hand, semantics are the main focus of ontologies. As such, the semantics in ontologies are explicit, maintainable, and up to date. For the reasons mentioned above, this research focuses on the ontology-based semantic data integration.

1.3 Formal Treatment of Conceptualization and Ontology

For the purpose of this research, Ontology is defined as specification of a conceptualization. Conceptualization is about concepts. It is an abstraction that consists of the relevant *concepts* and the *conceptual relations* that exist in a certain domain. A *concept* is a mental representation that picks out a set of entities, or a category. And *conceptual relations* do not depend on the existence of concrete instances in order to be true. Formal treatment of conceptualization is essential and a fundamental aspect of knowledge representation, Ontologies and information engineering. This is because, conceptualization is essential for the formalization of knowledge. There have been several attempts to formally model a conceptualization (Genesereth and Nilsson 2012), (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009). Some of these models are extensional models while others are of an extensional reduction nature. Because of the intensional nature of a conceptualization, it is very important that the conceptualization be modeled intensionally. Chapter 3 discusses the formal treatment of conceptualization and ontology in more details. It also proposes an intensional model for describing conceptualization and ontology that overcomes the limitations of the extensional and extensional reduction models. Since open environment is the main subject of this research, Chapter 3 shows that an intensional description of conceptualization and ontology is more appropriate for describing systems in open environment.

1.4 Ontology Mapping

Because of the heterogeneous nature of open environments, the use of ontology can help overcoming the heterogeneity issues. This is because ontologies provide explicit semantics about an information system. However, it is not to forget that even ontologies can be designed and maintained by different entities or individuals. As such, ontologies designed for the same domain can be heterogeneous. And so, a mechanism is required in order to bridge the heterogeneity gap that is an inherent in the definition of open environment. There needs to be a way to align these various ontologies and facilitate the interaction between them. This can take place by mapping the concepts of one ontology to the concepts of another ontology. The mapping process usually takes place after the components of one ontology are matched to the components of another ontology. After the matching takes place, the two ontologies can be aligned. The matching and alignment processes will make use of the semantics provided by the ontologies. This is one of the reasons why the use of ontologies is more powerful than the use of an information system schema. This is because the semantics provided by a database schema are not explicit and are usually outdated. On the other hand, an ontology is all about semantics and provides explicit semantics that are maintainable which can help making sure that the semantics are easily attainable, accurate, and up to date. When we say “semantic data integration”, we are referring to the fact that the mapping between various data sources makes use of semantics. This mapping is essential for the interoperability and interaction between various agents or information systems. Several frameworks and techniques are addressing the issue of ontology mapping, matching, and alignment (Bouquet et al. 2003), (Bouquet, Serafini, and Zanobini 2003), (Silva and Rocha 2003), (Maedche et al. 2003), (Besana, Robertson, and Rovatsos 2005), and (Giunchiglia, Yatskevich, and Shvaiko 2007). These techniques are critically reviewed in Chapter 2 and a conclusion is reached as which type of ontology mapping mechanisms is more appropriate for open environment.

1.5 Research Issues and Objectives

1.5.1 Formal modeling of Conceptualization and Ontology

The work investigates intensional logic, extensional logic and semantic integration principles to provide an intensional formal model for conceptualization and ontology. This formal model should be the base for the integration of various heterogeneous information systems. This integration will be derived by the mapping between the explicit ontological views of these information systems.

1.5.2 Surveying the Ontology Matching Algorithms

Various mapping algorithms have different characteristics and are suitable for different settings for data integration systems. This work surveys the structural and elementary algorithms for ontology matching in order to provide a matching algorithm to support discovering a rich set of semantic relations between various ontologies. The mapping algorithm should support the integration of various ontologies in open environment.

1.5.3 Modeling the Semantic Data Integration Framework in Open Environment

A pure intensional framework for ontology driven semantic integration should be ultimately developed. This framework will be empowered by the Formal Model and the Matching Algorithm. The proposed framework also should extend and improve the previously proposed solutions. Under this framework, the user should be provided with information residing in different data repositories. The user should be able to query against one ontology whereas, there answers will be calculated from various data sources in the environment. These data sources are assumed to be developed by different parties for the same domain. However, because the domain is fixed, the conceptual integration is possible.

1.5.4 Addressing the Dynamic and Loosely-Coupled Nature of Open Environment

In open environment, there is no centralized control. There is also no control over the set of participating entities or the number of participating entities. Each entity does not necessarily have knowledge of all other entities in the environment, rather each entity will have knowledge of the entities for which it has direct access. The proposed model needs to address the characteristics of open environment including the loosely-coupled nature and the dynamic nature of open environment.

1.6 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 reviews previous work proposed in the area of Data Integration and related topics. Chapter 3 proposes the formal intensional account for conceptualization, ontological commitment, and ontology. Chapter 4 presents the Intensional Model for data integration in open environments. Chapter 5 provides a Case Study analysis for the proposed intensional model. And finally, Chapter 6 concludes the theses and highlights some of the open issues that need to be addressed in future work.

Chapter 2

2 Literature Review

Data integration is considered to be the backbone for collaborative intelligence. It is essential for data mining, decision support, forecast, and business management. The subject of data integration has recently gained a lot of focus. And several approaches models, and architectures have been proposed in order to perform data integration. This chapter will present previous work that has been done in the field of data integration. More focus will be given to semantic integration, and in particular ontology-driven approaches especially the approaches that address open environment.

2.1 Data Integration

Data integration is the process of combining data residing at different sources, and providing the user with a unified view of these data (Lenzerini 2002). There are different paradigms and forms for data integration with different approaches and architectures. Some approaches perform the data integration task at the data level; others carry out the task of data integration at the conceptual level. Also, while some data integration approaches are physical in nature, other approaches are virtual in nature. (Bertossi 2007) Summarized some of the data integration forms as follows:

- *Materialized*: materialized data integration is physical in nature. In this form of data integration a physical, integrated repository is created, usually called a data warehouses. A data warehouse is a physical repository of selected data which is extracted from a set of databases and/or other information sources. Data are usually extracted from the data source, undergo some transformation process, and then loaded to the data warehouse through a process called known as Extract Transfer Load or ETL. As the naming suggests, in this type of data integration there has to be a mechanism to update the data warehouse, as soon as the data sources are updated, in order to maintain the integrity of the data. This form of integration is more suited for enterprise data integration or data sources that are created by the same agent. In this form of data integration, knowledge about the structures of the data and the technologies used to build each data source need to be available in order to facilitate the task of mapping and integration.

- *Mediated*: This form of integration does not extract the data from the data source. However, the data integration system makes use of the structure of the data or the semantics of the data to create a virtual unified view of the data. With the use of a global schema or an ontology that acts as a mediator, a virtual integration system is created. The mediator facilitates the interaction between various information systems while the data stays at the sources.

- *Federated and cooperative*: This form of data integration aim to integrate multiple distributed, heterogeneous, autonomous, database management systems or DBMSs. It maps a group of databases into a federated database by trying to create a balance between autonomy and information sharing. The group of federated databases in the federation is coordinated to collaborate.

- *Data Exchange*: This is a simple form of data integration in which data that is structured under one schema, called the source schema, are taken and transformed into data structured under the destination schema. In this form of integration, the actual data is taken and restructured. As such, there can be data loss in the data exchange process.

- *Peer-to-Peer data exchange*: the Peer-to-Peer or P2P data integration form is another form of virtual data integration in which data stays at the sources. The main difference between this from and the mediated data integration is the absence of a mediator. As such, several peers can exchange data without the need for a central control mechanism. Data is usually passed from peer to peer upon request, as query answers. In this sense, each peer is acting as a data integration system or a DIS on its own.

As mentioned above, some of these various techniques for data integration are *physical* and other approaches are of a *virtual* nature. When we say "*physical"*, we mean that the data is physically transferred from one data source to a data destination, a central repository or a data warehouse. During this transfer process, the data can undergo some transformation or restructuring as needed. And queries are usually answered by one single system or repository that holds all the integrated data. On the other hand, "*virtual*" data integration means that the actual data does not get transferred from one location to another. The data stays at its original source, and extracted upon request, i.e. as a query is being answered. As such, there must be a mechanism to map data sources to one another. In open environment, there are no constraints on the set of data sources or the number of data sources. The assumption is that, the system will allow data sources to enter and leave the environment at any point of time. As such, the data at each data source can be available to the system while the data source is part of the system. When a data source leaves the system in open environment, the data of such data source becomes unavailable to the system. As such, *Physical* data integration techniques not suited for such type of environment. And so, they will be out of scope of this work. Below, data integration approaches that are of *virtual* nature are described in more details.

2.1.1 Federated Data Integration

Federated Database Systems (FDBS) map multiple autonomous database systems into a federated database. An architecture for federated database systems in office information environment is proposed in (Heimbigner and McLeod 1985). The architecture proposed in (Heimbigner and McLeod 1985) aims to minimize the central authority of participating, possibly, autonomous data base systems while supporting partial sharing and coordination between database systems. The authors used the term federation to refer to the collection of constituent databases participating in a federated database. According to (Heimbigner and McLeod 1985) , the federation consists of the participating components and a single dictionary. The dictionary maintains the topology of the federation and keeps track of new components that enter the federation. The authors in (Heimbigner and McLeod 1985) argue that the federated database system should make a balance between autonomy and information sharing. This is because the entities participating in the federation need to maintain as much autonomy as possible while being able to share and receive information with other participating components. For this reason, the federated architecture need to support these two conflicting requirement of the federation. Four aspects of autonomy and three aspects of information sharing are discussed in (Heimbigner and McLeod 1985).

The four aspects of autonomy are as follows: the first aspect is that a participating component of the federation must not be forced to perform an activity for another component. Because a centralized authority overrides autonomy, a centralized authority cannot be a solution in a FDBS. Instead, there should be cooperative activities between components and supporting protocols need to be implemented. The second aspect of autonomy is that, each database should be able to determine the data to be available to other databases in the federation. That is to say, each database decides what data it wishes to share with others. Not only this, but also each component of the federation should be able to decide what components are allowed to access the information that is made available to the federation. The third aspect of autonomy is that, each component database should be able to determines how it will view and combine existing data. In other words, there should not be a single global schema, as is the case with composite systems, which is dictated by the federation. Instead each database should build its own global schema that is best suited for its needs. The forth aspect of autonomy is what the authors call “freedom of association”. The freedom of association is what maintains the dynamicity of the federation. The freedom of association allows each participating component to be able to enter or leave the federation at any point of time. Not only this, but also the participating components should have the authority to change their shared data interface by sharing new data or removing access to previously shared data.

As for the information sharing aspect of the federation, the authors, as mentioned earlier, discussed three ways for communication between data sources that the federation needs to support. These three ways of communication are summarized as follows:

1- Data communication: Components of the federation may be interested in accessing portion of the data that are owned by other components. As such, sharing information is an essential activity of the federation. And so, there need to be a way to share information in the federation.

2- Transaction sharing: In some cases, components of the federation may not want to share the row data; instead it may wish to share a processed version of the data. For this purpose, the federation must allow a mean by wish components are able to share transaction over the data instead of sharing the data directly.

3- Cooperative activities: Cooperative activities in this context refer to the ability of a component to negotiate data sharing with other components. Because the components of the federation are autonomous, cooperation is an essential requirement for the federation to be able to function correctly.

The architecture proposed in (Heimbigner and McLeod 1985) assumes homogeneous federation. That is to say, all the component databases of the federation have the same model. The model used is object-oriented database model. The federated database model in (Heimbigner and McLeod 1985) is based on three data modeling primitives. These three modeling primitives are namely; Objects, Types, and Maps. These three primitives are defined by the authors as follows:

- *Object*: An Object corresponds to a real world entity or a concept. Objects are divided into two categories: descriptor objects and abstract objects. As for descriptor objects, they are atomic strings of characters, integers, or Booleans, and generally serve as symbolic identifiers in the database. Non-descriptor objects are abstract objects. They are not directly displayable, except in terms of related descriptor objects (such as unique identifiers).

- *Types*: Types are time-varying collections of objects that share common properties; the objects of a given type are called the instances of that type. Some types are designated descriptor types in that they may only contain descriptor objects. All other types are designated abstract types. A type maybe a subtype of another parent type if it is defined so that its set of instances is always a subset of the instances of the parent type. Associated with any subtype is a predicate that determines which objects that are instances of the parent type are also instances of the subtype.

- Maps: Maps are "functions" that map objects from some domain type to sets of objects in the power set of some range type. A number of simple integrity constraints may be specified with each 'map; for example, a map may be specified to be single-valued (i.e., its value for all objects in the domain type has cardinality of zero or one) or multi-valued, and a map can be declared to be a unique identifier (key).

Each component database of the federation has three different schemas associated with it. These schemas are described as follows:

- ***Private Schema***: describes the portion of the component's data that is local to that component. Some of this information will remain local to the component itself, other parts will be exported to other components. The private schema also contains information and transactions relevant to the component's participation in the federation. This information is exported by other components. This contains descriptive information about the component and its export and import schemas.

- ***Export Schema***: describes the information the component is willing to share with other components.

- ***Import Schema***: specifies information that the component would like to use from other components.

In order to access an object from another component, the importing component needs to request access to the type of this object from the exporting component. After the access is granted, the importing component can add this type to its import schema and will have access to the objects of this type. This happens through a negotiation mechanism and after the request is granted, the importing component can access the objects of the imported type without further negotiation.

It is clear from the above discussion that the proposed architecture in (Heimbigner and McLeod 1985) assumes a homogeneous set of data bases in the sense that they are all built based on the same model. Not only this, but also the proposed model is very specific to a single database model; the object-oriented database model. In addition to this, the types are defined as a set of objects. This reflects the extensional nature of the proposed architecture. This can also be noticed from the mapping which is defined to be a function on the objects and not the types. As can be noticed, the proposed architecture can be applied in a closed environment or enterprise information environment. The proposed model in (Heimbigner and McLeod 1985) does not, however, address the issues of an open environment.

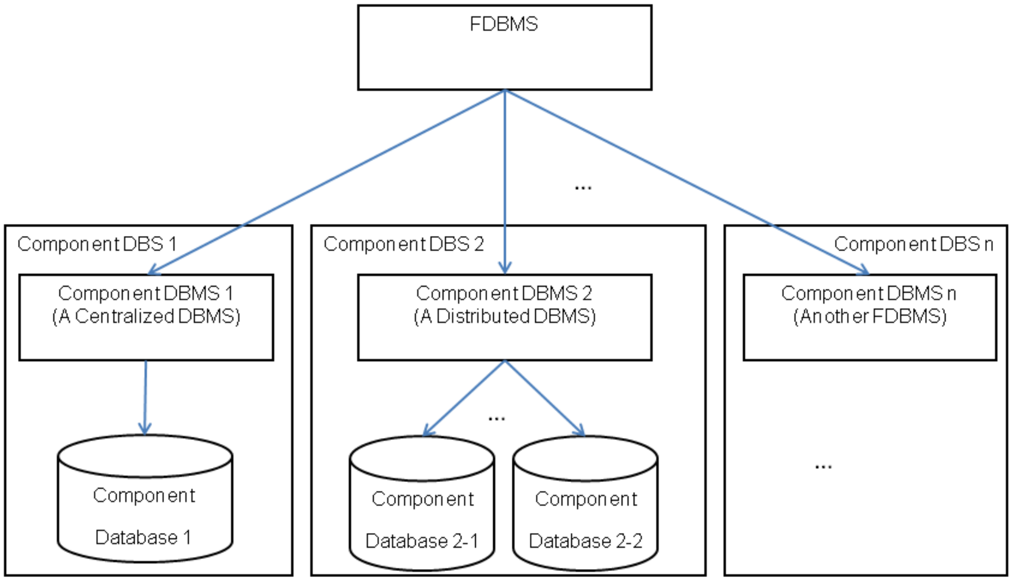

Figure 1: An FDBS and its components (Sheth and Larson 1990)

(Sheth and Larson 1990) defined the federated database system to be a collection of cooperating but autonomous, and may be heterogeneous component Databases. Like the database system has management system (DBMS), the authors in (Sheth and Larson 1990) called the coordinator of the component databases, Federated Database Management System (FDBMS). The authors called out three main characteristics of a FDBS. The three characteristics are namely; distribution, heterogeneity, and autonomy. These three characteristics are described as follows:

- *Distribution*: Data may be distributed among multiple DBSs in different ways, horizontal, or vertical. In D-DBSs, distribution may be induced to seek the benefits of distribution, which are increased availability, increased reliability, and improved access time. In FDBSs, much of the distribution is due to the existence of multiple DBSs before the FDBS exists.

- *Heterogeneity*: Heterogeneity can be due to differences in DBMS, or they can be due to differences in semantics of Data. The Difference in DBMS can be due to representational aspects or use different languages to manipulate the data. Representational aspects can be divided into difference in structure or support different types of constraints. An example for the difference in structure is the tables in the relational model vs the record type in an object oriented database system. Semantic heterogeneity can happen due to disagreement on the meaning, interpretation, or the intended use of same or related data.

- *Autonomy*: DBSs are always autonomous. The autonomy can be classified into: Design, communication, execution, and association autonomy. The design autonomy refers to the ability of a component database to choose its own design. And this is the primary reason for heterogeneity in FDBS. Communication autonomy gives the component DBS the right to decide whether to communicate with other component DBMS. The execution autonomy preserves the right of every component DBMS to execute local operations. This means, the FDBMS cannot enforce an order of execution of the commands on a component DBMS. The association autonomy requires that each component DBMS decides how much to share its functionality with others.

According to the authors in (Sheth and Larson 1990), a multi database system (MDBS) supports operations on multiple component DBS each of which is managed by different DBMS. MDBS can be federated or non-federated based on the autonomy of the component DBSs. In that sense, a FDBS is a compromise between, no integration and total integration. And so, FDBS support both local and global operations. But a FDBS users, cannot access local DBSs directly, rather they can access them through global operation. The component DBSs, though, should not differentiate between local and global operations. If it is the user’s responsibility to create and maintain the federation, an FDBS is said to be loosely coupled. However, in tightly coupled FDBS, the federation and its administrators, have the responsibility for creating the federation and controlling the access to all component DBSs.

As shown in Figure 1, the authors in (Sheth and Larson 1990) proposed a general architecture for FDBS which consists of some components; namely: Data, Database, Commands, Processors, Schemas, and Mappings. The authors defined different types of processors and schemas in their reference architecture for FDBS. There are four different types of processors defined in (Sheth and Larson 1990) as follows:

- *Transforming processors*: Translate commands from one language to another (command translation) or translate data from one form to another (data transformation).

- *Filtering Processors*: Contain the commands and associated data that can be passed to other processors. They only allow commands and data conversions that do not violate these filters.
- *Constructing Processors*: Are used for partitions and/or replication of an operation submitted by a single processor into operations that can be accepted by two or more processors. It also merges data produced by more than one processor into a single dataset that can be consumed by a single processor.

- *Accessing Processors*: Accept commands and produce data by executing the commands against the database

The authors in (Sheth and Larson 1990) describe a five-level schema architecture that extends the three-level architecture in the centralized DBMSs. The five-level schema architecture is defined as follows:

- *Local Schema*: the local schema is the conceptual schema of the component DBS.

- *Component Schema*: derived by translating the local schema into the data model called the common data model, or CDM for short, of the federated schema. They are used to facilitate negotiation and integration.

- *Export Schema*: represent the portion of the component schema that is available to the FDBS. It facilitates and manages the association autonomy. A filtering processor can be used to provide the access control as specified in the export schema.

- *Federated Schema*: is an integration of a multiple export schemas. The constructing processors translate commands on the federated schema into commands on one or more export schemas. There might be more than one federated schemas in FDBS, one for each class of federation users.

- External Schema: schema for a user or an application or a class of users or applications. Reasons for the external schemas are: customization (as the federated schema is very big and complicated), additional integrity constraints, and Access control (just as the export schema provide access control for data managed by component database).

As discussed in (Sheth and Larson 1990) there are two different approaches to build the federation. When the component DBS exist and it is required to integrate them, a bottom-up approach is used. On the other hand, if the FDBS already exist, and it is required to extend it to add a new component database, a top-down process is used. Below is a brief description of each approach:

- *Bottom-up approach*: This methodology is used to integrate several existing databases. This process involves, translate schemas to a CDM, define export schemas from a component schema, integrate schemas, and define external schemas.

- *Top-down approach*: This methodology is used when an FDBS already exists and additional user requirements are required. This process involves, defining or modifying external schemas, analyzing schemas by federated schemas to the external schemas, and integrating schemas. While doing the analysis step, the parts of the external schemas that are not supported by the federated schemas are captured in a temporary schema.

The authors in (Sheth and Larson 1990) also defined four major tasks for developing the federation, namely; schema translation, access control, negotiation, and schema integration. These four tasks are explained below:

- *Schema translation*: Schema represented in one data model is mapped into another schema represented in different data model. This is needed in two situations: translating a local schema into a component schema, and translating part of the federated schema into an external schema when the external schema is expressed in data model different than the CDM.

- *Access Control*: An FDBS should be designed to control access to component database by federation user. The system architecture has filtering processors at two levels, each of which can provide access control. The filtering processor between external and federated schemas control access to component DBSs. Likewise, the filtering processor between the external and federated schemas control access to federated schemas. Negotiation between component and federation DBSs may be necessary to reach an agreement on how to control the data a component database want to keep secured from some federated users while allowing access to other users.

- *Negotiation*: A federation DBAs and a component DBAs must reach an agreement about the contents of the contents of the export schemas such that federated schemas can be defined over them to support federation users. This dialog is called negotiation and follows certain protocols to govern message exchange.

- *Schema Integration*: Unlike view integration which refers to integration multiple user views into a single schema, schema integration integrates multiple database schemas into a single schema. It is divided into five steps. These steps are: pre-integration, comparison, conformation, merging, and restructuring.

It is quite evident that the approach proposed in (Sheth and Larson 1990) is based on schema matching. As mentioned earlier, even though schemas can provide us with some semantics, the main focus of database schemas is the structure of data and not the semantics. As so, the semantics in a database schema are not maintainable, and thus, are lost or outdated. That is why applying this approach to open environment can yield very poor matching quality even if we assume that the integrated databases are addressing the same domain.

2.1.2 Mediator-Based Data integration

A mediator-based data integration system usually consists of a global schema or ontology and a set of data sources (Lenzerini 2002). A general architecture for a mediator-based data integration system is shown in Figure 2. The mediator-schema is considered as a virtual data source. That means, data does not physically reside in the mediator-schema, rather, the mediator-schema serves as a unified schema for the integrated data sources. The data sources, on the other hand, contain the real data. And as such, there should be a way to determine the relation between the sources and the global schema. This relation is usually described as a mapping or interface between the local data sources and the global schema or ontology. Mediated data integration systems have gained a lot of attention in the last few decades.

A mediator-based data integration system is proposed in (Chawathe et al. 1994) and (Garcia-Molina et al. 1997). The proposed system is called TSIMMIS which stands for The Stanford IBM Manager for Multiple Information Sources. The proposed approach aims to help enterprises make decisions based on the integration of structured and unstructured data. A description of the main architecture of the system proposed in (Chawathe et al. 1994) can be seen in Figure 3. As can be seen in Figure 3, there is a collection of information sources, each of them is connected to a translator, or a wrapper. The translator's job is to convert the data objects to a common model. This happens by converting the queries of the common data model into a query over data source. This way, each data source can understand the query and execute it. After the query is executed, the translator converts the resulting data set, or the answer of the query, into the common data model.

It can also be noticed in Figure 3 that there are mediators above the translators. The job of the mediator in this approach is to direct the queries, merge the resulting answers to the queries, and also carry out some refinement process on those results. In that sense, the input to the mediator is a query written in terms of the common model, and the output is a data set in terms of the common model as well.

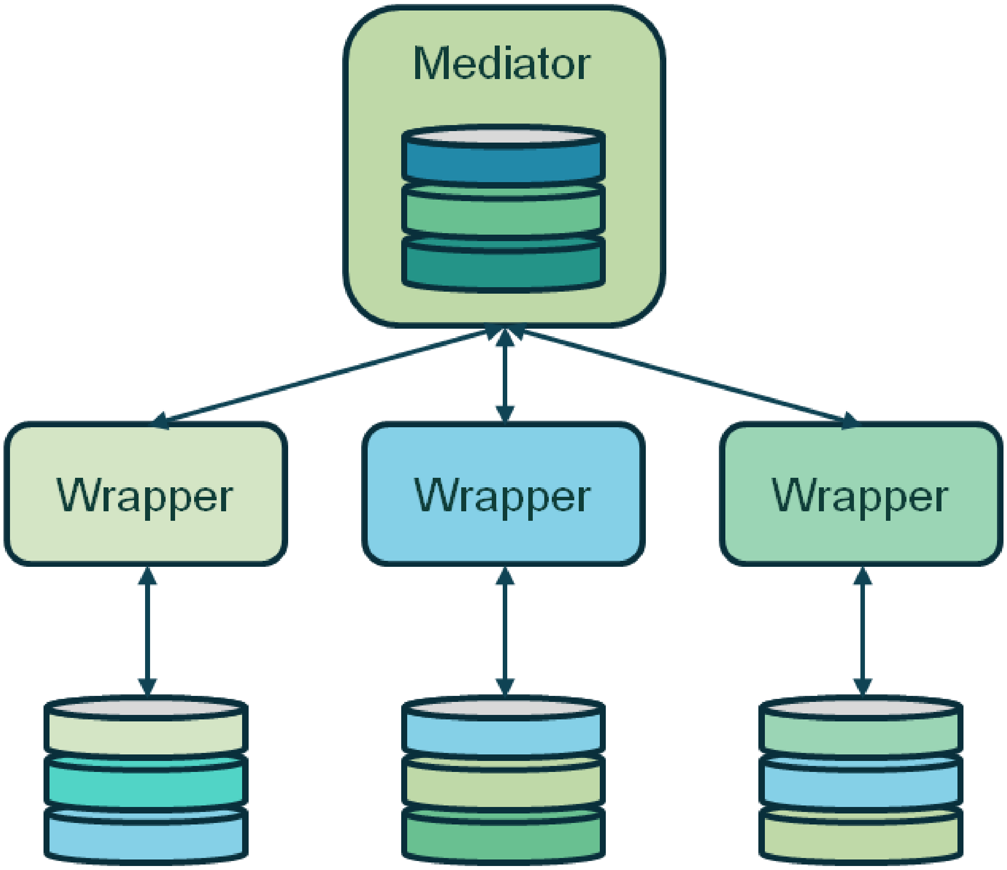

Figure 2: General architecture for mediated data integration system

Another component of the TSIMMIS approach is called the constraint manager. The constraint manager manages integrity constraints in order to guarantees the integrity of the data set returned from various data sources.

In (Levy, Rajaraman, and Ordille 1996), another mediator-based data integration system is proposed in order to integrate several relational databases. The proposed model also uses some object-oriented features in order to describe and reason about the contents of the relational data sources. In order to avoid modifying the global schema very often, the authors describe the data source objects as views or a query over the global schema. This makes the query answering very difficult since the mapping associates to each object of the source a query or a view over the global schema. As such, it is not straightforward to realize how to use each source in order to answer queries that are expressed in terms of the global schema.

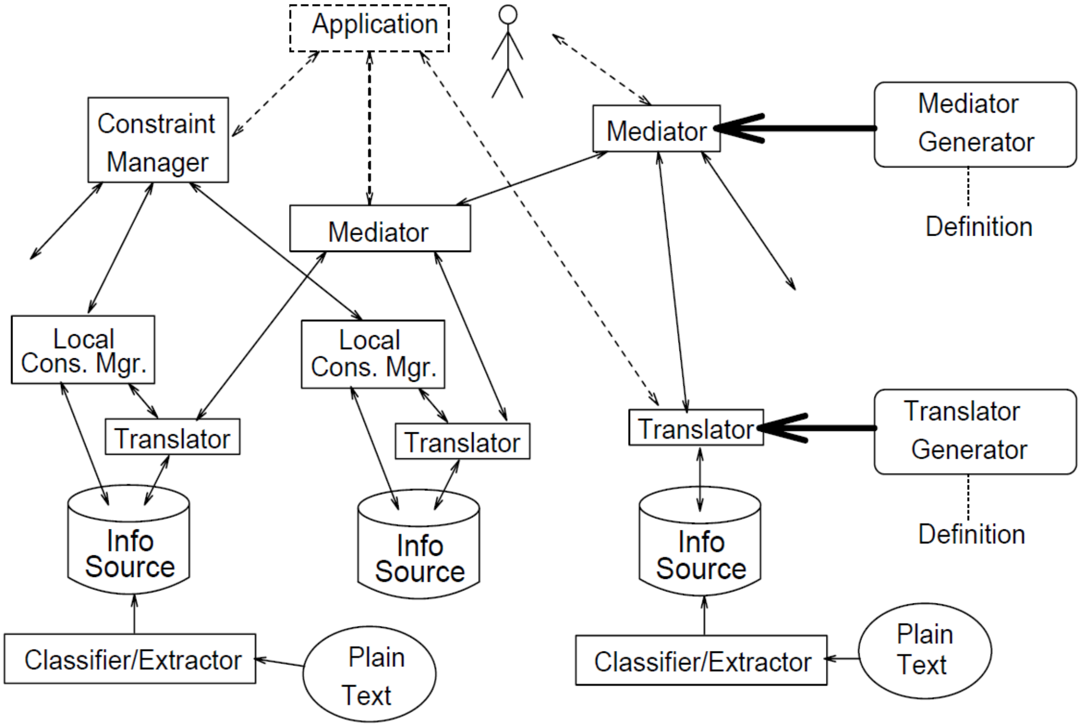

Figure 3: The TSIMMIS architecture (Chawathe et al. 1994)

Another system called Tukwila is proposed in (Ives et al. 1999). This system, which is displayed in Figure 4 focuses on the optimization of the query answering process through the proposal of a mechanism for query answering. The authors tried to address issues including, the absence of statistics about the data, unpredictable data arrival statistics, and overlap and redundancy among sources. The authors addressed these issues through the design of an adaptive technique. According to the authors, the Tukwila system is adaptive at two levels. While the first level is between the optimizer and the execution engine, the second level is within the execution engine itself. As shown in Figure 4, the main architecture of the Tukwila consists of the following components:

- *Query*: a query written in terms of the mediated relational schema.
- *Data Source Catalog*: The catalog contains metadata about the data sources participating in the data integration system.
- *Query Reformulation*: the query reformulation process takes as an input the user query and produces a union of queries that refer to the various data sources.
- *Query Optimizer*: Transforms the reformed queries into an execution plan for the execution engine.
- *Query Execution Engine*: The query execution engine executes the plan produced by the query optimizer.
- *Wrappers*: The wrappers facilitate the communication between the query execution engine and the data sources. They also translate the data from the form used by the data source schemas to the format of the mediator or global schema.

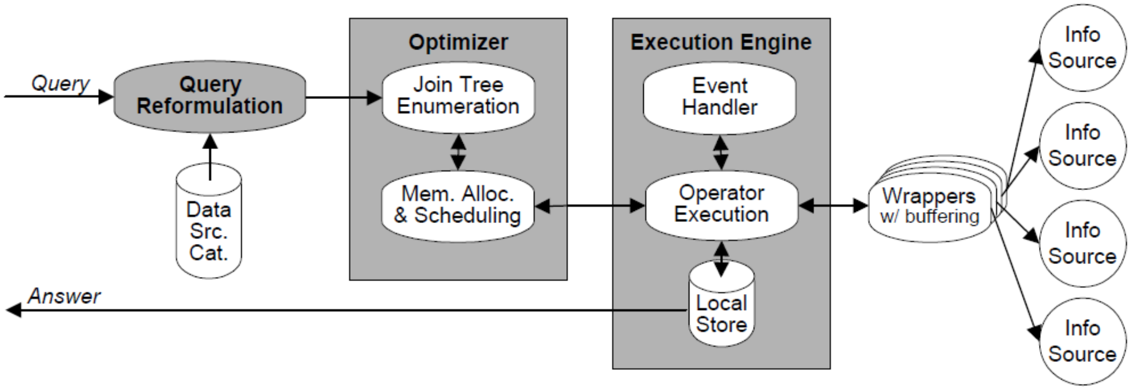

Figure 4: The Tukwila architecture (Ives et al. 1999)

It can be inferred from the previous description that the Tukwila system is using a traditional mediated based architecture in which there is a mediator, a global schema, set of sources with local schemas, and a set of wrappers. This comes with some enhancement in the query processing and execution. It is also clear that the set of data sources share the same model and as such are homogeneous in that sense. It can also be seen that the model that is used in (Ives et al. 1999) is a relational database model.

(Lambrecht, Kambhampati, and Gnanaprakasam 1999) and (Kambhampati et al. 2004) proposed another optimization algorithm that makes use of heuristics that guide a greedy optimization algorithm. While the use of a greedy minimization algorithm optimizes the data gathering plan by removing redundant and overlapping data sources, the use of heuristics guides the greedy minimization algorithm to remove costlier information sources first. The authors used a traditional mediated-based data integration algorithm with an improved query answering technique. Another thing to note is that, the authors in (Lambrecht, Kambhampati, and Gnanaprakasam 1999) use a LAV approach in which the objects of the source schemas are described as views over the global schema. As mentioned earlier, this approach is challenging when it comes to query answering. The problem of rewriting a user query expressed over the global, mediator, schema to a query over the source schemas becomes a problem of answering queries using views. The LAV approach in general has the advantage of not modifying the global schemas when a new source is added to the data integration system. However, as explained above, the query answering process becomes very challenging with the use of a LAV approach.

2.1.3 Peer-to-Peer Data integration

As described in the previous section, the mediator-based architecture requires the existence of a centralized control in the form of a mediator that is connected to all the data sources in the data integration system. On the contrary, a Peer-to-Peer P2P architecture does not require the existence of a centralized control. Rather, each data source connects to other data sources in the network and exchange queries and answers without the involvement of a mediator. In that sense, each data source acts as a data integration system on its own. As such, each peer exports data in terms of its own schema, and data interoperation is accomplished through mappings between the schemas of these peers (Calvanese et al. 2004). While the mediated data integration systems architecture is centralized by nature, the P2P data integration systems adopt a completely decentralized approach (Calvanese et al. 2004). This can be seen when comparing the network in Figure 5 to the one in Figure 2. In this section, some of the data integration techniques that are based on the P2P architecture are reviewed.

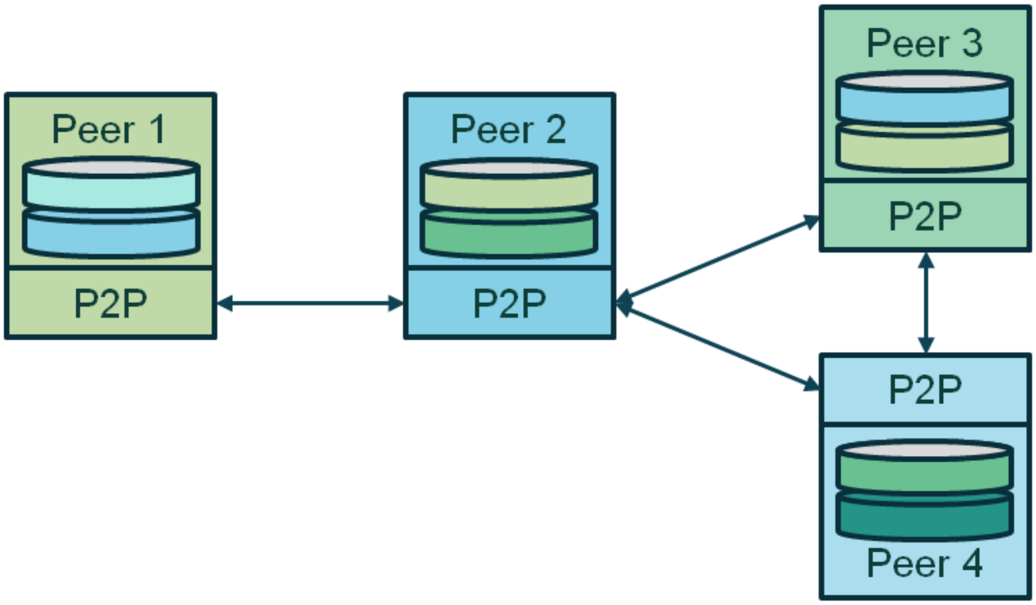

Figure 5: General architecture of a P2P Network

P2P architecture was introduced in file sharing systems. Several system that use P2P in file sharing can be found in (“Freenet” n.d.), (“LOCKSS” n.d.), and (Yang and Garcia-Molina 2002). (Ng, Ooi, and Tan 2002) introduced a generic P2P system called BestPeer in order to serve as a platform on which P2P applications can be implemented. The BestPeer network consists of two types of different entities the first type of entities is the node. Here nodes represent computing entities and there can be a large number of nodes in the network. The second entity is a location independent global name lookup or what is referred to as (LIGLO). Each participating node in the P2P network must run the BestPeer through which the node, or computational entity, can share resources with other participating peers in the BestPeer network. The platform introduced in (Ng, Ooi, and Tan 2002) integrates two main technologies; namely mobile agent and P2P architecture. The main purpose of using P2P architecture in BestPeer platform is to facilitate resource sharing amongst participating peers in the network. The authors also employed mobile agent technology in order to further extend these functionalities. According to the authors, the use of mobile agents enables the peer to share more than just files; rather peers can use mobile agents in order to collect processed information such as summaries or even collect statistics on the entire P2P network. Peers can also share information on a coarse-granularity or fine-granularity level. Here the coarse-granularity level may refer to sharing an entire file, for example. On the other hand, a fine-granularity level may refer to the partial sharing of a file. Another interesting attribute of the platform proposed in (Ng, Ooi, and Tan 2002) is that, it enables the sharing of resources, or computational power. For example, if a peer is requesting a file from another peer, the requesting peer can send an algorithm along with the request. The peer that is providing the information can run the algorithm on the file before sending a response to the requesting peer. As such, not only the requesting peer acquires information from the peer that is providing the information, but also the requesting peer used the computational capabilities of the peer that is providing the information. The algorithm sent with the request can be some sort of filtering or any processing on the requested information. The Independent Global Names Lookup Server (LIGLO) mentioned above provides each peer with a unique global identity. This server is itself a node in the network that has a fixed IP address and is running special software to serve its purpose. The server generates what is called a BestPeer Global Identity or (BPID) for each participating peer in the P2P network. It also keeps track of each peer's current status such as IP address and whether the peer is currently online or offline and so on.

In order for the P2P architecture to be used for data management, (Daswani, Garcia-Molina, and Yang 2003) as mentioned in (Calvanese et al. 2004), suggested several requirements including; Autonomy, Expressiveness of query language, Efficiency, Quality of service, and Security. For more details about these requirements, we refer the reader to (Daswani, Garcia-Molina, and Yang 2003) and (Calvanese et al. 2004).

(Ng et al. 2003) proposed a P2P distributed data management system which supports context-based search. The system introduced in (Ng et al. 2003) is called PeerDB and is a database application that is implemented on top of BestPeer (Ng, Ooi, and Tan 2002) . More precisely, the authors implemented a SQL database system op top of the BestPeer network. Each node, in the system proposed in (Ng et al. 2003) consists of four main components. Namely; the data management system, the database agent system, the cash manager, and the user interface. These four components are displayed in Figure 6.

The first main component of the node in a PeerDB network is the data management system. The main job of the data management system is to facilitate storage, manipulation and retrieval of the data at the node. The authors use MySQL Database as a storage server. Thus, the system can be used on its own as a standalone DBMS outside of the P2P network. There are also two sub-components associated with this component. These components are called the local dictionary and the export dictionary. Since the PeerDP is P2P a relational database management system, the local dictionary stores metadata associated with the relations in the database. On the other hand, the export dictionary stores the metadata of the objects that are made global in the network. As such, the other nodes in the network won't have access to all the relations in the relational database schema. Rather, only objects that are exported are made sharable to other nodes in the P2P network. In that sense, the metadata associated with the export dictionary is a subset of the metadata in the local dictionary.

The second main component of a PeerDB node is the database agent system or DBAgent. This component provides the environment for mobile agents to operate on. Each node in the network has an agent called the master agent. The master agent manages the query of the user, clone and dispatch agents to neighboring nodes in the P2P network, receive answers to the queries, and present the answers to the user. The master agent also manages what is called reconfiguration policies and monitors the statistics about the node.

The last two are the component of a PeerDB node are the cash manager and the user interface. The cash manager cashes remote data in a secondary storage that is local to the node. It also determines the policies for cashing and replacement. On the other hand, the user interface provides a user-friendly environment for user to submit user queries, maintain sharable objects, and insert/delete objects in the database.

In (Kementsietsidis, Arenas, and Miller 2003), the authors discuss the issue of data mapping between heterogeneous data sources residing on various peers in a P2P network. As the authors described, in a file sharing system where there is no heterogeneity, searching for a file on a P2P network is not very challenging. The search usually takes place using a file name or the name of an album in a music sharing system for example. This is because these songs or albums have agreed on names that are homogeneous across all peers in a P2P network. The issues discussed in the work proposed in (Kementsietsidis, Arenas, and Miller 2003) are mainly focused on domain where such common agreements do not exist. Such systems are heterogeneous in nature and each peer may have its own naming convention for its own files. Since each node, peer, can have its own local applications that depend on the local naming convention at the node, it is not realistic to force a global naming convention for the files residing at different nodes in the P2P network. As such, in order to be able to search data in such heterogeneous environments, traditionally, mapping tables are employed. These mapping tables store the correspondence between values. The simplest form of a mapping table is a binary table that contains pairs of corresponding identifiers from two different peers. Those mapping tables represent expert knowledge and are usually created and maintained by domain experts. Because the manual creation and maintaining of the mapping table can be a very expensive process, the authors in (Kementsietsidis, Arenas, and Miller 2003) present alternatives semantics for the mapping tables and a language that allow the specification of the mapping tables under different semantics. The authors proposed the treatment of mapping tables as constraints that can be reasoned about on the exchange of information between various peers in the P2P network. The reasoning will help inferring new mapping constraints or check if a set of mapping constraints is consistent.

(Halevy et al. 2003), (Halevy et al. 2004) and (Taylor and Ives 2006) proposed P2P techniques for information sharing in which peers publish their data on an ad hoc basis. In (Halevy et al. 2004), the authors proposed a P2P data management system called PDMS for the integration of several relational databases. The proposed system takes advantage of the HTML web and the semantics of the data management applications. The system consists of a set of data sources. Every single data source is represented by a peer in the P2P network. Each peer defines its own relational peer schema and various peers are related to one another through a set of mappings. Users can place a query to any of the peers in terms of the relational schema of the peer. The main focus of the work presented in (Halevy et al. 2004) is to allow for scalable data integration system as opposed to mediated data integration systems in which the mediator schema may be required to be updated with the addition of new data sources.

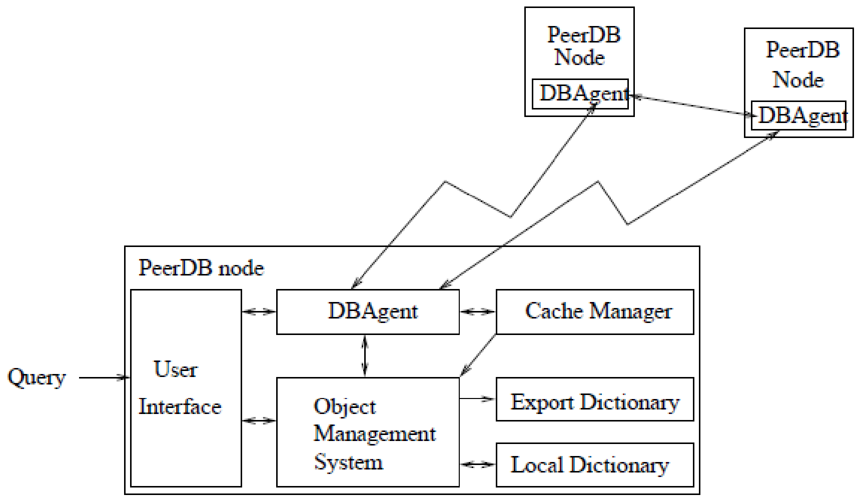

Figure 6: PeerDB node architecture (Ng et al. 2003)

Another issue in the data integration systems is querying the various data sources in the network. In (Huebsch et al. 2005) the authors developed a general purpose relational query engine for relational P2P data integration system. The proposed query processor targets the very large scale P2P data integration networks of thousands or even millions of nodes on the internet. The proposed query engine adopts a relational data model in which data values are fundamentally independent of their physical location in the network. In order to achieve a high level of scalability, the authors used distributed hash table in order to provide location independent naming and network routing. The execution environment of the proposed query engine consists of a virtual runtime interface and an event-handler. The virtual runtime interface encapsulates the basic execution platform. On the other hand, the multiprogramming is achieved via an event-based programming model running on a single thread.

The authors in (Milo et al. 2005) address the issue of guiding the materialization of intensional data in XML documents. In order to understand the problem we quote the following paragraph from (Milo et al. 2005).

Intensional data is provided by programming constructs embedded inside documents. Upon request, all the code is evaluated and replaced by its result to obtain a fully materialized HTML or XML document, which is then sent. In other terms, only extensional data is exchanged. This simple scenario has recently changed due to the emergence of standards for Web services such as SOAP, WSDL and UDDI. Web services are becoming the standard means to access, describe and advertise valuable, dynamic, up-to-date sources of information over the Web. Recent frameworks such as Active XML, but also Macromedia MX and Apache Jelly started allowing for the definition of intensional data, by embedding calls to Web services inside documents. Since Web services can essentially be called from everywhere on the Web, one does not need to materialize all the intensional data before sending a document. Instead, a more flexible data exchange paradigm is possible, where the sender sends an intensional document, and gives the receiver the freedom to materialize the data if and when needed. In general, one can use a hybrid approach, where some data is materialized by the sender before the document is sent, and some by the receiver.

A benefit that can be seen immediately is that the user can get some information, like the local weather forecast just by activating the corresponding service call, without having to reload the whole document. The authors then used an ActiveXML P2P news exchange system to implement their approach.

In (Adjiman et al. 2006), the authors are interested in P2P inference systems in which each peer can answer queries by reasoning from its local theory but also can ask queries to some other peers to which it is semantically related. In doing so, each peer needs to have some partial knowledge about some other peers in the network. As such, when a peer is asked to perform a reasoning task, if the peer cannot solve the task completely on its own, using its own local knowledge, the peer will distribute some reasoning subtasks among other peers in the P2P network. The output of all the subtasks must then be recomposed in order to construct the output to the initial task. The authors then applied their algorithm to reasoning in the semantic web settings.

In (Lumineau, Doucet, and Gançarski 2006), the authors proposed a system in which each node in the P2P network can represent a peer or a set of peers. When a node represents a set of peers it is called a super peer. The super peer is a node that can represent a company, for example, with each computer or user in the company representing a regular peer. This technique is used in order to enhance the accessibility between peers in a network that has a very large number of nodes.

(Yang and Garcia-Molina 2002) discussed three different techniques to optimize search in a P2P network. These techniques are namely; Iterative Deepening, Directed BFS, and Local Indices.

The iterative deepening initiates multiple breadth-first searches with successively larger depth limits. The search continues until either the query is satisfied, or the maximum depth is reached. This technique is good when satisfying the query is important. On the other hand, when minimizing the response time is more important, the Direct BFS is recommended in (Yang and Garcia-Molina 2002) to be the choice. In Direct BFS search technique, the queries are sent directly to a subset of nodes that are expected to yield many results in a short period of time. The Local Indices search technique however aims at maintaining the satisfaction while keeping the search cost low at the same time.

For more comprehensive review of several P2P techniques, the reader is referred to the survey in (Androutsellis-Theotokis and Spinellis 2004).

2.2 Semantic Data Integration

In order for two individual to interact successfully, there are explicit or implicit assumptions that they share the same semantics about the subject of interaction. Without the shared semantics, the interaction is very likely to be unsuccessful. One of the goals of information integration is to support interoperability among information systems. This is why it is important to be able to tell when various statements are about the same subject. If the different information systems use the same model and representation language to describe the domain, the integration task would be easy. But, when information systems use different representation languages and/or different models, the use of semantics is very important. The semantics used for data integration can be taken from the schemas of the information systems or can be carried out by ontologies.

(Hull and King 1987) discusses the importance of the semantic database modeling and described their view of the generic components of the semantic database model. The authors also emphasized the need for a higher level modeling abstraction and the reduction of the semantic overloading of the data type constructs. Recent research has pointed out that, even though database schema contains semantics, database schemas are mainly concerned about data and data structures. Moreover, the semantics in the database schema are hardwired, lost, tossed, or out of date (Uschold 2015). Furthermore, the semantics in the database schema are un-maintainable since they are implicit. On the other hand, the main focus of ontologies is not the structure of some data. Rather, it is the meanings and description of the conceptualizations, and subject matters. Ontologies also provide explicit semantics and are maintainable. These are main reasons why ontologies have gained acceptance as sources of semantics. Also, given that in open environments data is not always structured, the need for explicit semantics becomes quite evident for DI in open environment. Going forward, the focus of this work will be on ontology-driven data integration.

2.2.1 Schema-Based Data Integration

Database schema integration is the process of integrating the schemas of existing databases into a global unified schema. The schema-based data integration has been in the community for longer time than the ontology-based data integration. Even though both techniques can depend on semantics, the main focus of the schema-based data integration is the structure of the data. This is because the inherent nature of a data schema which cares mainly about the structure of the data. A study conducted in (Batini, Lenzerini, and Navathe 1986) offers a unifying framework for the problem of schema integration. The authors also provide a comparative analysis of other methods done in the field of the schema-based data integration. The database integration produces a global schema of a collection of databases. The global schema is a virtual view of all database schemas taken together in a distributed system. According to the authors, any schema integration technique can be considered a mixture of four main activities. These activities are:

1- Pre-integration: This activity conducts analysis on the schemas before integration in order to decide on some integration policies. This will control the order in which the integration takes place. It also governs whether to integrate the entire schema or a portion of the schema. For example, a preference can be given to a financial schema over a production schema and so on. Other factors that can be affected by the pre-integration activity is the amount of designer interaction and the number of the schemas to be integrated at one time. These decisions are also made during the pre-integration phase. Also, the collection of information relevant to the schema-integration task is considered as part of this phase. This information can be assertions and constraints among views, for example.

2- Comparison of the schemas: during this activity, the schemas are analyzed and compared in order to determine matching amongst concepts and detect any potential conflict that may exist.

3- Conforming the schemas: once a conflict is detected during the previous activity, the real effort to resolve these conflicts is made during this activity. Resolving detected conflicts makes possible the merging of various schemas. According to the authors, automatic conflict resolution is generally not possible. As such, in any real-life integration activity, designers and users are required to interact with the system during this step.

4- Merging and restructuring: in this step, after the resolution of conflicts, the very task of merging the various schemas takes place. This results in a unified global integrated schema.

In (Spaccapietra, Parent, and Dupont 1992) however, the authors view the schema integration as a two phase process. In the first phase, commonalities and discrepancies among input schemas has to be found, the authors call this “the investigation phase”. The authors propose the usage of names, structures, and constraints in order to perform this phase automatically. This will require the confirmation of a DBA in the end to approve or deny the automatic findings. The second phase in the schema-integration process is what the authors called the schema integration. This phase is a semi-automatic phase that takes place based on the inter-schema correspondences and the integration rules. This phase is semi-automatic because it also requires the interaction of a DBA in order to resolve conflicts between input schemas every time the integrator does not have the knowledge to do it.

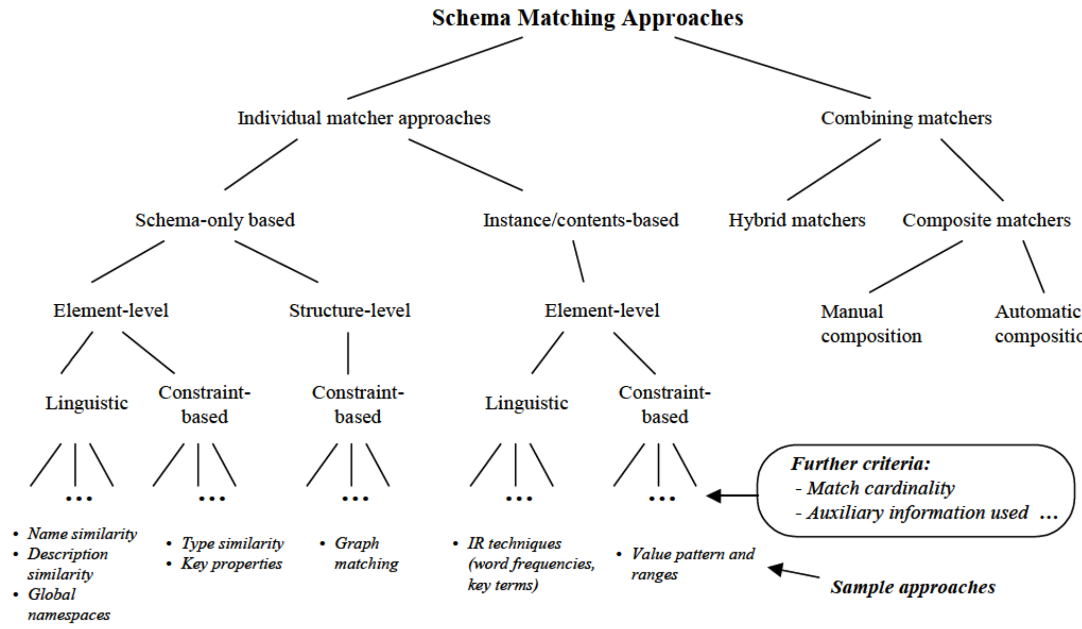

Figure 7: Classification of Schema Matching Approaches (Rahm and Bernstein 2001)

Finding the proper matching between heterogeneous database schemas is in the core of the task of integrating various database schemas. (Rahm and Bernstein 2001) conducted a survey on various techniques used to achieve the task of schema matching. The authors in (Rahm and Bernstein 2001) classified the schema matching approaches into two main categories the individual matcher approaches or the combining matcher approaches. Each of these main classifications is further classified into sub-classifications. This is partially captured in Figure 7. In their classifications the authors considered the following classification criteria:

1- Instance vs schema matching: This criteria looks into whether the matching technique takes advantage of the data contained in the database or perform the matching, only, based on schema-level information

2- Element vs structure matching: According to the authors, the matching can be performed for individual schema element, such as a relation or an attribute, or on the level of a combination of elements. As an example of a combination of elements will be a complex schema structure that can include more than one relation

3- Language vs constraint matching: The matching can take as an input the names and textual descriptions of schema elements. This type of matching is what is referred to here to be based on Language. In the contrary, the matching can ignore the names and textual description of the elements and focuses on the constraints, such as primary keys or foreign keys in the relations. This type of matching is what the authors refer to as constrained-based matching

4- Matching cardinality: according to the authors, the cardinality distinguish matching algorithm based on how the matching algorithm relate elements of one schema to the elements of another schema. For instance, the matching results may relate one or more elements of one schema to one or more elements of the other schema. This will yield one of four cases1:1 mapping, 1:n mapping, n:1 mapping, or n:m mapping.

5- Auxiliary information: This classification will distinguish matching algorithms according to the inputs they consume. According to the authors, most of the schema matching algorithms do not just take two schemas to match. Rather, there are more inputs that go along with the input schemas. These inputs can be dictionaries, global schemas, previous matching decisions, or the input of a user or an expert.

In (Mendling, de Laborda, and Zdun 2005) the authors discussed the issue of applying schema integration in order to integrate XML schemas for business process modeling. The authors classified the schema integration techniques into three main categories. These three categories are, manual schema integration, semi-automatic schema integration, and automatic schema integration. The manual schema integration leverages the knowledge of domain experts. On the other hand, the semi-automatic schema integration techniques rely on assertions to state semantic relationships between concepts of various schemas to be integrated. These assertions can be thought of as integration rules that are used by a so-called integrator to generate a unified global schema. Although this approach is less time-consuming, it also depends on the knowledge of domain experts or DBAs to state these assertions or integration rules. And finally, the automatic schema integration uses techniques from information retrieval and artificial intelligence to detect semantic relationships between elements of various database schemas. These techniques are less time consuming and they do not require the involvement of a domain expert or a DBA in the integration process. However, there is no guarantee that they yield results that are as accurate as the manual and semi-automatic techniques.

There is a lot of effort that has been spent in the field of schema-based data integration. The trend nowadays is, however, towards ontology-based semantic data integration. This is because the semantics in the database schema is really not the focus of the database schema. As such, there is a lot of hidden rules inside a database schema that makes it more appealing for application use than it is for querying. On the other hand, those hidden rules in a database schema are all made explicit in ontologies. This is because the main focus of ontologies is the semantics rather than the structure of the data. The following section will shed some light on ontology-based data integration techniques.

2.2.2 Ontology-Based Semantic Data Integration

Ontology based data integration has also gained a lot of attention in several fields including; medical fields (Kama et al. 2012), biology (Sütterlin et al. 2013), enterprise information systems (Song, Zacharewicz, and Chen 2013), document-oriented queries (Coletta et al. 2012), (Castanier et al. 2013), (Canito, Maio, and Silva 2013), virtual production (Reinhard et al. 2012), product development process (Woll, Geissler, and

Hakya 2013), quality assessment (J. Wang 2012) , toxicology (Boyles et al. 2019), air traffic management (Egami et al. 2020), and many more fields.

In (Coletta et al. 2012) and (Castanier et al. 2013) the authors developed an environment for real-life data integration scenarios over public data called WebSmatch. The work in (Coletta et al. 2012) and (Castanier et al. 2013) relies on an ontology matching and alignment algorithm called YAM++ (Ngo and Bellahsene 2012) The integration environment then consumes the matching output in clustering documents. The clustering aims to classify documents in several categories. The work in (Coletta et al. 2012) and (Castanier et al. 2013) used data integration and ontology matching to provide the users with recommendations regarding documents they may be interested in. WebSmatch did not follow a particular model for data integration; instead, it just relied on matching and then clustering of the data. The inputs to the application are preexisting databases of documents. The application then works in sequence to extract the metadata, match, and then cluster. The addition of a new data source would require the application to run again from the beginning and perform these three steps. That said, this algorithm assumes a closed environment, and cannot be applied to open environment. The reason is that, in open environment, entities need to enter or leave the environment at any time without making an effect on the behavior of the system.

In (Kama et al. 2012), the authors used full domain ontology, schema mapping, and reverse engineering mechanisms (D2RQ) (*The D2RQ Platform – Accessing Relational Databases as Virtual RDF Graphs* n.d.) in order to generate a Data Definition Ontology from database information. Doing so, the authors overcome the semantic gab that does exist between data sources when explicit semantics are not defined. However, the reverse engineering process will not generate accurate semantics, and cannot be fully automated. In open environment, the database designs can be very diverse that it is very hard to extract useful semantics from the database schema. The method proposed in (Kama et al. 2012) focuses on the generation of the DDOs and then defines some rules to align these ontologies. Since it relies on the schemas in order to generate the semantics, the work in (Kama et al. 2012) implicitly has a closed world assumption. The work in (Kama et al. 2012) also did not specify a model for the data integration system; instead it just relied on matching the generated DDOs. It is not clear how the queries are processed, and there does not seem to be a way to account for new data sources.

The authors in (Reinhard et al. 2012) use data integration techniques in virtual production process. Ontology based data integration techniques are used in (Reinhard et al. 2012) to resolve inconsistency between different specialized simulation tools and to exchange their resulting data. In (Hoehndorf et al. 2012) the authors employed phenotype ontologies to integrate phenotype descriptions within and across species. The authors relied on ontologies like Gene Ontology GO (Mungall et al. 2011) and a Phonotype Quality Ontology (PATO) (*Phenotype And Trait Ontology* n.d.) and implemented their axioms using OWL. In that work also, the existing data is used to drive the ontology. That said, the ontologies used are DDOs. DDOs are not suitable for open environment since the data schemas can be very diverse. Also, since there is no predefined ontology, this method must have implicit assumptions about the structure of the input data sources. These assumptions make it inappropriate to be generalized to open environment.

In the field of integrating enterprise data, the authors in (Song, Zacharewicz, and Chen 2013) proposed a semantic information layer (SIL). The SIL acts as mediation media among heterogeneous information systems to overcome gaps of data and semantic heterogeneity. The authors used reverse engineering to retrieve DDO from the relational databases and used some ontology alignment and matching techniques to generate mappings. The authors used the mappings between the recovered ontologies and the relational database systems to support query answering. The ontologies are recovered from the relational database schemas using reverse engineering. This reverse engineering process can result in a non-accurate semantics fed to the ontologies, and in turns the SIL layer. As mentioned earlier, using the schema to drive semantic is not appropriate for open environment since the structure of the data can vary dramatically. It is also worth mentioning that the authors in (Song, Zacharewicz, and Chen 2013) used a model for the DIS that is similar to the one in (Lenzerini 2002) . This model is an extensional model and does not account for the intensional nature of the open environment.

In (J. Wang 2012) a framework to measure the quality of Data Integration Systems is proposed. The proposed framework is ontology based and employed ontology reasoning in order to generate an integrated quality view of the Data Integration settings. The author also consumed some ontology matching algorithm to support the purpose of integration. This framework assumes preexisting data sources that are termed: ITEM, METRIC, QUALITY CRITERIA and USER. This is again an example of an extensional data integration system designed for certain problem with a closed world implicit assumption.

In (Sütterlin et al. 2013) the authors extend the EPISIM platform (Sütterlin et al. 2009) in order to allow direct integration between System Biology Markup Language (SBML) model and Cell Behavioral Model (CBM). The authors used semantics in order to integrate cellular states like proliferation and differentiation expressed in (CBM) represented by graphical process diagrams to biochemical reaction or gene regulatory networks expressed in (SBML). It is quite evident that this method cannot be generalized to open environment since it has assumptions about the structure of the data.

The authors in (Canito, Maio, and Silva 2013) used ontologies in order to describe document repositories. The authors call their method Ontology-Driven Data Cleaning and Enrichment ODCE. The algorithm in (Canito, Maio, and Silva 2013) is applied to the output of a Natural Language Parsing (NLP) process over a set of documents. As the authors in (Canito, Maio, and Silva 2013) described, the output of an NLP process is a set of facts represented in a lightweight ontology. The objective of the proposed method in (Canito, Maio, and Silva 2013) is to automatically integrate and enrich the output of the NLP process into a knowledge base whose contents are described in terms of a richer ontology that captures the same domain. In this method, the knowledge is physically merged as opposed to virtually integrated. This is not appropriate for an open environment in which the sources have autonomy and can have control over their knowledge in terms of answering queries or changing the contents. It is also worth mentioning that the users adopted the same model for data integration discussed in (Lenzerini 2002) which is extensional in nature.

In (Woll, Geissler, and Hakya 2013) the authors focused on integrating different ontologies that are developed for several stages in the Product Development Process PDP. The authors used ontology integration patterns in order to handle different relations between ontology concepts. These integration patterns are mainly used in order to avoid bloating ontologies with irrelevant concepts. The integration patterns used are: integration as extension, ‘shared high-level concepts, and hybrid integration which is a combination of the previous two patterns Figure 8. This framework is developed for an organization that has control over its data sources. The framework assumes preexisting ontologies for predefined data sources. These ontologies are: Project, Process, Requirements, Product, Design Rationale, and Optimization ontologies. This assumption is clearly a closed world assumption and the framework is design for a specific problem. That said; the framework may not generalize to open environment as it has a closed world assumption and is extensional in nature.

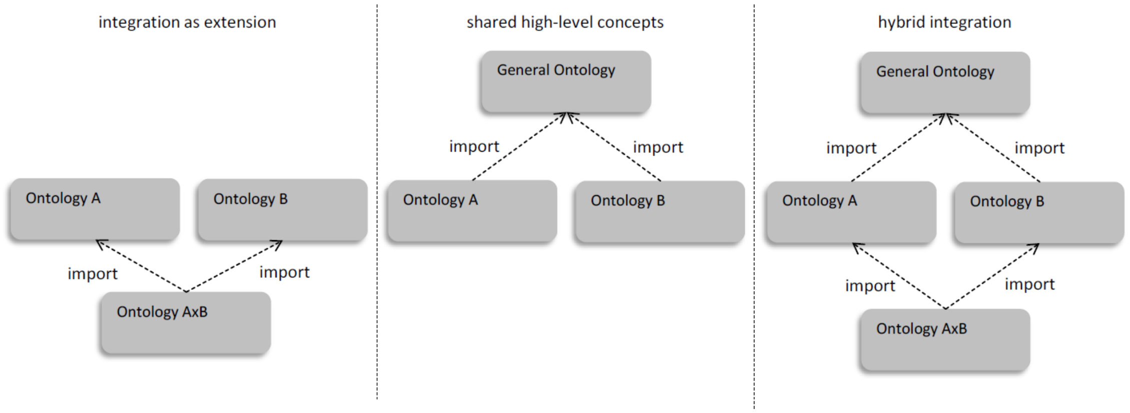

Figure 8: Ontology integration patterns for linking elements from different ontologies (Woll, Geissler, and Hakya 2013)

(Chen et al. 2017) used a goal driven learning process to construct an ontology that evolves through a learning process. In doing so, the authors in (Chen et al. 2017) used some Link Grammar Parse, and WordNet API in order to extract the semantics from the text. As such, the technique results in a data driven ontology (DDO). The focus of the work in (Chen et al. 2017) is the construction of a DDO rather than performing the data integration task. While it is important to drive ontology from the data when ontology is lacking, it is important to note that textual data is not a reliable source of semantics. This is because the semantics are implicit, subjective, and not maintainable.

In (Calvanese et al. 2018) and (De Giacomo et al. 2018) the authors present a general framework for ontology-based data access. The general architecture in (Calvanese et al. 2018) and (De Giacomo et al. 2018) consists of three main components, one ontology, a set of data sources, mappings. The system presented in (Calvanese et al. 2018) does not require each data source to have its own ontology. As such, the mapping is not between the ontology of the data source and the global ontology. Rather, the mapping in (Calvanese et al. 2018) is between the data that resides in the data sources and elements of the ontology. The authors in (Calvanese et al. 2018) used relational databases to wrap the data sources, a data federation tool, and descriptive logic for data access and query answering. For ontology-data mapping, the authors in (Calvanese et al. 2018) used a GAV approach. This is natural as the architecture used is mediated architecture.

(Ferreira et al. 2019) presented a collaborative environment which benefits from the implementation of an ontology-based data integration architecture to provide the user with an integrated view of the data. The proposed framework collects data from various sensors and then standardizes the collected information in terms of *subject-data-object* according to the Resource Description Framework (RDF) (Flesca et al. 2017). This is implemented using the Apache Jena Framework (Louie et al. 2007). The Apache Jena Framework is used to employ four ontologies to standardize the data based on their context. A semantic integrator is used to add semantics to the data. It also sorts the captured data to the mediation ontology. The mediation ontology integrates the data sources requested by the user and provides a global integrated view. The architecture provided is centralized in nature. It allows data to be collected from various sources and integrates them through a mediation ontology. This addresses the issue of heterogeneity between various data sources. However, it does not address the dynamic, distributed, or the loosely coupled nature of open environment.

In (Li et al. 2020), a framework for bridge health monitoring is proposed. The proposed framework presented a semantic model called Bridge Structure and Health Monitoring (BSHM) Ontology. The framework introduced in (Li et al. 2020) attempts to provide global semantic schema that is more expressive than database schemas. It also attempts to take advantage of the ontology-based reasoning to infer implicit knowledge as opposed to just relying on the knowledge stored in a database. The architecture of the system presented in (Li et al. 2020) has four main layers. The lower level layer is the data acquisition layer. The data is then mapped through a mapping engine in order to be stored in the data storage layer. The data storage layer does not have a specific model for data representation. Various models for data representation can be used, which are then unified through the use of a global ontology, the BSHM ontology. The user can query the data through the application layer. This layer then passes the query to the query and reasoning engine which represents the data access layer. The query and reasoning engine, in turns, executes the query against the unified ontology, which sits on top of the data storage layer to provide a unified view of the data. The framework proposed in (Li et al. 2020) addresses the issue of heterogeneity in the data structures. It offers a single unified global ontology in order to standardize the data stored using various data models. The framework in (Li et al. 2020), however does not address the semantic heterogeneity which can exist between ontologies of various information sources (Alkhamisi and Saleh 2020). It also has a centralized view and does not address the issues of open environment including; the distributed, loosely-coupled, or dynamic nature of open environment. It is also worth mentioning that, according to the authors in (Li et al. 2020), the scope of the proposed work is limited to bridge structure division, structural properties, management information, SHM systems, sensors and sensory data.

It is interesting that none of the work referenced above has provided a solution that addresses the data integration in open environment. We know that the use of explicit ontology helps bridging the gap between heterogeneous data sources. However, the use of ontology alone cannot address the problem of open environment in which data sources can enter or leave the environment at any time. As explained in (Alkhamisi and Saleh 2020), the ontology-based data integration systems that utilize a single unified global ontology do not support the addition or elimination of a data source from the system. If a data source is to be added or removed from the system, the global ontology has to change to adapt to the change. As such, it is important for the data integration system in open environment to be designed so that it supports the dynamic nature of open environment. The model also needs to address the loosely-coupled natures of open environment.

2.2.3 Ontology and Conceptualization

There is a debate between researchers about what an ontology is. Since the definition that is mostly cited in the information sharing community considers an ontology to be “a specification of conceptualization”, we find it useful to start by, informally, discussing what a conceptualization is. Then, we will revise the discussion about the term ontology.

2.2.3.1 Conceptualization

Different researchers have different definitions of what a conceptualization is. This depends on their field of interest and the model they use to describe the conceptualization. The authors in (Genesereth and Nilsson 2012) defined the conceptualization to be:

> “The objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them”

It can be noticed from the above definition that the definition includes objects (instances) and the relations between them. And that explains why the authors use extensional notation for conceptualization. In (Gruber 1993) however, the author adapted the following definition for a conceptualization:

> “An abstract, simplified view of the world that we wish to represent for some purpose”

Here, the motivation of the conceptualization is to serve the purpose of representation. And since the representation, of a piece of reality, requires abstraction of that piece of reality, it is natural to create a conceptualization for this reality first. Another definition can be found in (Borst 1999) which defines a conceptualization to be:

> “A structured interpretation of a part of the world that people use to think and communicate”

An intensional account of conceptualization is reflected by the definitions in (Guarino and Giaretta 1995), (Guarino, Oberle, and Staab 2009), and (Xue 2010). In (Guarino and Giaretta 1995) , the authors adopted the following definition for conceptualization:

> “An intentional semantic structure that encodes the implicit roles constraining the structure of a piece of reality”

And in (Guarino, Oberle, and Staab 2009) a conceptualization is considered to be implicit. To the authors; implicit means in the minds of peoples. And this is why it needs to be explicitly specified through an ontology. And finally, in (Xue 2010) a conceptualization is defined as:

> “An abstract model that consists of the relevant concepts and the relationships that exist in a certain domain”

The last two definitions reflect the intensional nature of conceptualization. And that is why the authors in (Guarino and Giaretta 1995) and (Guarino, Oberle, and Staab 2009) emphasized that conceptualization is about meanings. And as such, conceptualization should not change unless meanings do change.

2.2.3.2 Ontology

The definition of ontology is still debatable in the information sharing research community. Ontology is first defined in (Gruber 1992), (Gruber 1993), and (Gruber 1995) as follows:

> “Ontology is an explicit specification of a shared conceptualization”.

It is also argued in (Smith and Welty 2001) that the definition of Gruber in (Gruber 1992) and (Gruber 1993) is very broad and allows for too many interpretations. (Guarino and Giaretta 1995) and (Guarino 1997) criticized the previous definition for being relying on an extensional notation. The extensional notation causes the conceptualization, and in turns the ontology, to capture dynamic knowledge about the domain while ontology is supposed to capture only static knowledge (Borst 1999). Ontology is then defined in (Guarino and Giaretta 1995) to be:

> “Ontology is a logical theory which gives an explicit, partial account of conceptualization”

Another definition that was adopted for ontology in the same article (Guarino and Giaretta 1995) is:

> “Ontology is a synonym of conceptualization”

(Guarino and Giaretta 1995) also interprets the term explicit as a concrete symbolic level object. Having this understanding in mind, the two definitions adopted in (Guarino and Giaretta 1995) are very different in nature. While the former considers ontology to be some concrete symbolic theory, the later definition is far away from any representational considerations. In (Borst 1999) , the author defined ontology as:

> “Ontology is a formal specification of a shared conceptualization”

Emphasizing that, there must be agreement on the conceptualization that is specified. This is because; the ontology may not be reusable if there is no agreement on the conceptualization it specifies. In (Studer, Benjamins, and Fensel 1998) the authors combined the two definitions in (Gruber 1993) and (Borst 1999) as follows:

> “Ontology is a formal explicit specification of a shared conceptualization”

This definition emphasizes the two sides which are; the explicitness and the formality. Ontology needs to carry explicit semantics as opposed to implicit semantics extracted from data structures (i.e. schemas). Ontology also needs to be represented in a language that is machine readable. And that is why it was emphasized that it is a formal specification.

2.2.4 Ontology and Knowledge Representation

Because a well-defined syntax, formal semantics, and an efficient reasoning support are crucial for high quality ontology, the study of various knowledge representation and reasoning techniques is in the core of representation and formal treatment of ontology. In this section we discuss some knowledge representation techniques. Our study will be motivated and derived by the need for ontology to be represented in a formal language. This formal language need to be expressive enough to allow us to represent whatever facts in the domain of discourse. The language also needs to provide powerful reasoning tools that can efficiently infer implicit facts, answer queries, and perform other reasoning tasks. Speaking about ontology representation; it is important to understand what a representation mean. A representation is defined in (Brachman and Levesque 2004) to be a relationship between two domains in which the first is meant to stand for or take place of the second. Usually the first, representor, is more concrete, immediate and more accessible than the other.

For a machine to be able to understand and reason about knowledge, this knowledge needs to be expressed in a formal way to avoid ambiguity and vagueness. Knowledge representation then is defined in (Brachman and Levesque 2004) as, the field of study concerned with using formal symbols to a collection of propositions believed by some agent. It is argued that, not all the believed propositions need to be represented. Only part of the believed proposition will be represented, and it is the job of reasoning to bridge the gap between what is believed and what is represented. Instead of literally defining what a knowledge representation is, the authors in (Davis, Shrobe, and Szolovits 1993) discussed the roles that a knowledge representation plays. As argued by the authors, a knowledge representation plays five main rules. These rules create demands that are, sometimes, conflicting. These demands in turns lead to a set of properties the representation is required to have. The rules of knowledge representation as mentioned in (Davis, Shrobe, and Szolovits 1993) are:

- *Knowledge representation is a surrogate*. Most of the things that we want to reason about do exist in the real world. The representation works as a surrogate, for those things, inside the Knowledgebase and the reasoner. The correspondence between the surrogate and its referent in the real world is the semantics of the representation. Since representations usually create some simplifying assumptions and artifacts, it is important to consider how close the representation to the things it represent. The authors argue that, any representation of the real world will be imperfect. And as such, the quality of the decision taken by the reasoners will depend on how good the representations approximate its referents.

- *Knowledge representation is a set of ontological commitments*. Knowledge representation is a set of ontological commitment in the sense that, in choosing a representation technology, one is bringing certain aspects of the world into focus. This should be based on the understanding of what parts, of the world, are relevant, and what aspects are less relevant or irrelevant. To make it easy to understand, the authors mentioned an example for choosing between Logic and Frames. In choosing Logic, a minimal commitment is being made about seeing the world in terms of individual entities and the relationships between them. On the other hand, choosing a frame-based technology has us thinking of classes, class hierarchies and instances of classes (objects).

- *Knowledge representation is a fragmentary theory of intelligent reasoning*. It is theory because it is believed that knowledge representation is motivated by human reasoning. But, since it only reflects parts of the belief that motivated it, this is why it is fragmentary. The definition of intelligent reasoning will vary depending on the field by which a representation is inspired. For example, for views derived from mathematical logic, intelligent reasoning is some variety of formal calculations. Other views, rooted in psychology, see intelligent reasoning as a characteristic human behavior. Based on the nature of the conception of the representation, some kinds of inferences are said to be legal, or supported by this representation. Finally, we need to know the recommended inferences because what we can infer is not necessarily what we should infer. Also, since the set of legal inferences is sometimes very large, the set of recommended inferences help making the reasoning intelligent.

- *Knowledge representation is a medium for efficient computations*. If we think mathematically, reasoning is a computational process, and this process needs to be done efficiently. In the field of knowledge representation and reasoning there is always a tradeoff between the requirements of the Representation, and the requirements of Reasoning. The representation prefers the language to be expressive enough to be able to represent whatever facts we want to represent about our domain. On the other hand, the more expressive the language is, the less efficient the reasoning is performed.

- *Knowledge representation is a medium for human expression*. So, not only it needs to be expressive, but also it should be easy to use. Here what matters is not, what we can use the language to express, instead it is, how easy it is to use the language to express something.

The authors in (Davis, Shrobe, and Szolovits 1993) pointed out that knowledge representation is not just a Data structure, it does have semantics which interprets its symbols and constructs and relate them to meaning in the real world. Also, each representation has certain characteristics that facilitate the use of the language in the way it is intended to do, rather than what it can do. This aspect implies that, the appropriate way of using a representation is using it in its intended spirit rather than getting the language to do something that is not ordinary though it is capable of doing. Another important point is that, selecting a representation technique means choosing a conception of the fundamental nature of intelligent reasoning. This is because knowledge representation techniques differ in conceptual aspects, rather than implementation aspects.

In the following sections, we will try to use these rules to examine various knowledge representations techniques for the purpose of representing ontologies.

2.2.4.1 Propositional Logic

Propositional logic is logical language with the simplest semantics. It is a quantifier free language, and in turns contains no bound variables. If you strip First Order predicate

Logic from quantification, it yields Propositional Logic. Below, the syntax and semantics of propositional logic is illustrated. Then, representing ontology in propositional logic is discussed.

The syntax of propositional logic consists of the following:

- A countable set of propositional symbols (atoms).
- The logical connectives: ∧ (and), ∨ (or), ⊃ (implication), ¬ (not).
- A body is an atom or is of the form b1∧ b2, b1∨b2, ¬b1 where b1 and b2 are bodies.
- A definite clause is an atom, or is a rule of the form b⊃ h where h is an atom and b is a body.
- A knowledge base is a set of definite clauses.

As for the semantics propositional logic, it is known that semantics relate the formal symbols in the logic to things in the domain you are representing. In propositional logic the following semantics hold:

- An interpretation "I" assigns a truth value to any assertion of a proposition.
- A body b1 ∧ b2 is true in "I" if b1 is true in "I" and b2 is true in "I". Otherwise it is false.
- A body b1 ∨ b2 is true in "I" if b1 is true in "I" or b2 is true in "I" Otherwise it is false.
- A body ¬b1 is true in "I" if b1 is false in "I". Otherwise it is false.
- A rule b⊃ h is false in "I" if b is true in "I" and h is false in "I". Otherwise it is true.

Although propositional logic has very clear and well-defined semantics, and further, it takes advantage of the Boolean nature of efficient reasoning, it is not a good candidate for representing ontology. This is because the language has a very limited expressivity, and also the kinds of inferences allowed by the language are very limited. This means that ontology represented in propositional logic will be limited in both the expressivity and the implicit facts that can be inferred from the represented facts. It is also worth mentioning that, the propositional logic does not have an explicit model for the relationship between concepts in the represented domain. And so, even the clear semantics of the propositional logic does not provide a good representation of ontology. This is because ontology sees the domain of discourse as concepts and relationships that exist between these concepts.

2.2.4.2 Frame Language

Inspired by psychology which sees intelligent reasoning as a characteristic human behavior, (Minsky 1974) argued that, when one encounters a new situation, one selects some structure from the memory. This structure is called a frame. And it is framework that one remembers and, if required, may be adopted to fit the current situation. So, according to (Minsky 1974) a frame is a data structure that represents a stereotyped situation, and that has some relevant information attached to it.

The concepts of frame have evolved over time to reflect certain understanding of the nature of representation in the context of knowledge representation and reasoning. Frame is a way of representing knowledge in an object-oriented manner. Derived from the object-oriented nature of the frames, frames contain named lists of slots in which fillers can be placed. While the slots in a frame represent the properties or attributes of the frame, the fillers are the values of these properties or attributes. There are two types of frames, *individual frames*, and *generic frames*, and they represent objects and classes, respectively, from the object oriented point of view. The syntax of an individual frame looks like the following:

```
(Frame-name
        <slotName1   filter1>
        <slotName2   filter2>
…)
```

The fillers can either be atomic values or names of other individual frames. Individual frames also have a special slot called INSTANCE-OF. The filler of this frame is a generic frame's name determining the class of the individual frame. The following example is presented in (Brachman and Levesque 2004):

```
(Toronto
        <INSTANCE-OF     CanadianCity>
        <Province        Ontario>
        <Population      4.5M>
…)
```

Instead of the INSTANCE-OF slot in the individual frames, the generic frames have an IS-A slot which is filled by a more general frame. The following example is also presented in (Brachman and Levesque 2004):

```
(CanadianCity
        <:IS-A        City>
        <:Province    CanadianProvince>
        <:Country     canada

)
```

You can see that, the generic frame CanadianCity is a specialization of the more generic frame City. The generic frames’ slots can also have special procedures as fillers. These procedures are prefixed by IF-ADDED, IF-NEEDED, or IF-REMOVED, and they are executed if a value of the slot being added, needed, or removed, respectively.

As can be noted from above, frame language supports inheritance. The IS-A slots arrange the generic frames into a taxonomic structure. They pass the propertied of the parent frame down to its children (specializations). A generic frame can have more than one parent in the hierarchy. In that case, this generic frame is considered to be a specialization for all its parents. And it inherits all the properties of its parents. The usage of the INSTANCE-OF slot then passes the properties of the generic slot down to its instances (individual frames). Both of these processes, passing the properties from the parent frames to their specializations and from them down to their instances, are recognized as inheritance of properties. The inherited properties can be overridden by other fillers (values) for the slots (properties). And this is what makes the frame system described as defeasible.

If there are slots in a generic frame that are filled with procedures prefixed by IF-NEEDED, the values of these slots are not calculated, for the instances or specializations of this generic frame, till they are required. Other procedures, prefixed by IF-ADDED, are executed only when a value is entered, or calculated, for this slot. And the procedures prefixed by IF-REMOVED are executed only when a value for that slot is emptied.

As for reasoning, reasoning with frames involves both matching a certain object or situation to a frame, and applying general information to specific instances. The reasoning procedure can be summarized in the following loop:

a) When the user declares the existence of a situation or an object, this object is matched to the best applicable frame.
b) An instance of this frame is then created. And the values of its slots, if empty, are inherited.
c) For each slot of that instance that has filler, if there is any IF-ADDED procedure that can be inherited is executed.

d) If the execution of the IF-ADDED procedure results in the instantiation of a new frame, the algorithm jumps to Step b).

In case if the user, or external system, requires the filler for a slot, the algorithm proceeds as follows:

a) If there is a filler stored in the slot, it is returned.
b) Otherwise, if there is a filler that can be inherited, its value is returned.
c) Otherwise, if there is any IF-NEEDED procedure that can be inherited, it is executed.
d) Otherwise, the value of the slot remains unknown.

As for representing ontology with frames, Frame-based systems are more organized as compared to the flat nature of logic-programming. Using frame-based systems knowledge is expressed in more structured way and the encapsulation property of the object oriented paradigm is taken advantage of (Trentelman 2009). When logic-programming is used, the information about one entity can be scattered among a number of seemingly unrelated sentences in the knowledge base (Brachman and Levesque 2004) . But, frame-based systems group all the information about certain entity in one structure. Frame-based systems also offer efficient means for decidable reasoning (Trentelman 2009). Moreover, the storage of knowledge in a dynamic fashion that can be calculated and modified during run time is a further advantage of the frame-based systems. In spite of the advantages of the Frame-based systems, they have some drawbacks for which they may not be the perfect fit for ontology representation task. The main focus of frame-based languages is to represent the domain in terms of objects and classes, and capturing the taxonomic structure of the class hierarchy. Although this is a conceptual advantage, the expressive power of the language is still limited, i.e. capturing the relations between objects is not a primary focus of the frame-based systems. Another example that shows the limited expressivity of the frame-based systems is the difficulty to represent heuristic knowledge. For example, it is easy to express the following facts in FOL:

Ali was married to Madiha during the period (1970, 1975).
Then he got married to Laila in the period (1977, 1985).

While the same task is not that easy in frame-based systems. It is also worth mentioning that, the semantics of frame-based systems are not precisely defined. This problem is addressed in (Selman and Levesque 1993). And this is another strong reason for which frame-based systems are not the perfect choice for expressing ontology which require a representation with well-defined semantics.

2.2.4.3 Description Logic

Description Logic (DL) is a family of description languages extends both semantic networks (Richens 1956) and frames. It also adds formal logic semantics to these representation techniques (Trentelman 2009). It represents the basis for Web Ontology Language OWL which is commonly used and accepted in the semantic web. In this section, the following will be discussed: the syntax of the DL, the semantics of DL, reasoning with DL, the relationship between DL and FOL, and finally, representing ontology with DL will be examined.

1) *The syntax of DL* :

   The vocabulary of DL supports three types of non-logical symbols, namely:

   a. Constants: which represent named entities in the domain of discourse. constants are usually written in uppercase.

   b. Atomic concepts: which represent concepts in the domain of discourse. These concepts can be thought of as the types or class of the entities that the constants represent.

   c. Atomic roles: which represent binary relationships that exist between entities in the modeled domain, usually written in a camelCase style, with the first letter being small.

   The non-logic symbols are common between all types of description languages. The DL also has two special atomic concepts ⊤ (top) and ⊥ (bottom). The logic symbols used along with the non-logical symbols determine the type of the description language. The various types of the description languages are different in their degree of expressivity and, in turns, in the complexity of reasoning with each language. The description logic language that is the least expressive is the Attributive Language $\mathcal{AL}$ which features the following symbols:

- The constructors $\sqcap$ (conjunction): which is interpreted as set intersection
- The negation $\neg$: which is interpreted as set complement
- The universal (value) restrictor $\forall$ and the existential (value) restrictor $\exists$
- The right and left parenthesis and the comma
- The constructors $\sqsubseteq$ which is interpreted as subsumption.
- The equality $\doteq$.

If we use R to range over roles, C to range over the concepts, and $\mathcal{A}$ to range over atomic concepts, then the allowed concepts in $\mathcal{AL}$ are the following:

1. Every atomic concept is a concept.
2. If $C1$ and $C2$ are concepts, then $C1 \sqcap C2$ is a concept.
3. If A is an atomic concept then $\neg A$ is a concept.
4. If R is a role and C is a concept then $R.C$ is a concept.
5. If R is a role and $\top$ is the top concept, then $\exists R.\top$ is a concept.
6. Nothing else is a concept.

This language does not allow role constructors, and hence, all roles are atomic.

$\mathcal{ALC}$ extends $\mathcal{AL}$ where the $\mathcal{C}$ stands for complex complements. In this presentation, C does not have to be an atomic concept in order for $\neg C$ to be a concept. One can add more expressiveness power to the language by allowing more complex constructs. As mentioned earlier, there are many varieties of DL that offer different degree of expressiveness. Table 1 lists some of these expressivities and their labels. For more details on the different DL languages and extensions, we refer the reader to (Baader et al. 2007)

In (Trentelman 2009), five types of syntactic expressions were mentioned. These expressions are; concepts, roles, constants, assertions, and terminological axioms. Terminological axioms specify how concepts are related to each other. For example, if *C1* and *C2* are concepts, then *C1* ⊑ *C2* and *C1* ≐ *C2* are terminological axioms. While the former is interpreted as subsumption, i.e. all individuals who satisfy *C2* necessary satisfy *C2*, the later shows that the two concepts are equivalent. Equalities are referred to as definitions, and they are usually used to give symbolic names for complex expressions. A set of terminological axioms constitute a TBox. A knowledge base represented in DL consists of a two main components, the TBox and the ABox, for assertions. While the TBox contains a set of terminological axioms, the ABox is a set of assertions about individuals (constants).

Table 1: DL expressivities and their labels

Label	Expressivity Added
$\mathcal{C}$	Complex Complements
$\mathcal{U}$	Union constructs
$\mathcal{E}$	Full existential quantification
$\mathcal{N}$	Number restriction
$\mathcal{S}$	Abbreviation for ALC with transitive roles
$\mathcal{H}$	Role Hierarchy
$\mathcal{I}$	Inverse properties
$\mathcal{Q}$	Quantified cardinality restrictions

2) *The Semantics of DL*:

A Concept in DL denotes the set of all individuals in the domain satisfying the properties of this concept. And a role, on the other hand, represents a relationship between entities in the domain. In that sense, if C is a concept and R is a Relationship, then $\forall R.C$ represents individuals that are R related to only individuals of the class C. On the other hand, $\exists R.C$ represent the set of individuals who are R related to at least one individual of the class C. For example, if Cake is a concept, and eats is a relationship, then ∀eats.Cake represents the individuals who only eat Cake (Trentelman 2009). We can literally read it as “All they eat is Cake”. Also, the semantics of DL will vary depending on the degree of expressiveness and the allowed constructs. In this section, the semantics of the $\mathcal{ALUEN}$, as described in (Trentelman 2009), will be presented.

For the $\mathcal{ALUEN}$ language, a model is a pair (D, F) with D being the domain, and F is an interpretation function assigning to each element in the vocabulary of the $\mathcal{ALUEN}$ a semantic value in the domain. Each constant is interpreted as an entity in the domain. Each atomic concept is a set of domain entities. And each role is understood as a binary relationship, i.e. $F(R) \subseteq D \times D$. This is interpreted as the set of pairs of entities that are R related. Following are the interpretations of all the concepts allowed in $\mathcal{ALUEN}$ as listed in (Trentelman 2009).

For the distinguished top concept $\top$, $F(\top) \equiv D$.

For the distinguished bottom concept $\bot$, $F(\bot) \equiv \emptyset$.

- $F(C1 \sqcap C2) \equiv F(C1) \cap F(C2)$.
- $F(C1 \sqcap C2) \equiv F(C1) \cup F(C2)$.
- $F(\neg C) \equiv D \setminus F(C)$.
- $\mathrm{F}(\forall R.C) \equiv \{x \in D \mid \text{for any } y, \text{ if } (x, y) \in F(R) \text{ then } y \in F(C)\}$.

- $F(\exists R.C) \equiv \{x \in D \mid$ there is at least one y such that $(x, y) \in F(R)$ and $y \in F(C)\}$.
- $F(\leq nR) \equiv \{x \in D \mid$ the cardinality of $\{y \mid (x, y) \in F(R)\} \leq n\}$.
- $F(\geq nR) \equiv \{x \in \mathrm{D} \mid$ the cardinality of $\{y \mid (x, y) \in F(R)\} \geq n\}$.

Now the satisfaction which is expressed as $M \vDash \alpha$, and read as "α is satisfied in *M*", can be defined as follows:

- $M \vDash C$ iff $F(C) \not\equiv \emptyset$
- $M \vDash C1 \sqsubseteq C2$ iff $F(C1) \subseteq F(C2)$
- $M \vDash C1 \doteq C2$ iff $F(C1) \equiv F(C2)$
- $M \vDash C(a)$ iff $F(a) \in F(C)$
- $M \vDash R(a, b)$ iff $(F(a), F(b)) \in F(R)$

3) *Reasoning in Description Logic*:

The following inferences are provided by DL in order to deduce implicit knowledge from explicitly represented knowledge (Staab and Studer 2010).

- Subsumption: determines subconcept-superconcept relationship. A concept *C1* is subsumed by a concept *C2* if all instances of *C1* are necessarily instances of *C2*.
- Instance: determines whether an individual is an instance of a concept.
- Consistency: determines if a KB is not contradictory.
- Equivalence: determines whether two concepts are equivalent.
- Satisfiability: determines whether a concept is satisfiable with respect to a TBox.

Equivalence can be reduced to subsumption since two concepts are equivalent if they subsume each other. Also, subsumption can be reduced to satisfiability since $C \sqsubseteq_{\mathcal{T}} D$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$. Satisfiability and the instance problem can also be reduced to the consistency problem. In that sense, all the reasoning problems can be reduced to consistency problems, which can be solved using a consistency algorithm.

4) *Relation to FOL*:

Most DLs are decidable fragments of FOL (Staab and Studer 2010) in which role names can be seen as binary predicates, concept names can be viewed as unary predicates, and individual names may be perceived as constants. Also, the constructors $\sqcap$ and $\neg$ correspond to the FOL connection and negation respectively. Likewise, the restrictors ∀ and ∃ correspond to the FOL universal and existential quantifiers, respectively.

As mentioned earlier, high quality ontology requires both well-defined semantics, and efficient reasoning algorithm. And since description logic provides both, it can be a good candidate for representing ontology. To feel the expressiveness of DL, let us point out some features of the $\mathcal{SHIQ}$ language as presented in (Staab and Studer 2010) . Qualified number restrictions of the form $\leq nR.C$. Using this feature, not only one can specify that a person has at most two children, ($\leq 2 hasChild$), but also you can specify the type of these children ($\leq 2 hasChild.Female$) . $\mathcal{SHIQ}$ also allows for complex terminological axioms. For example, you can say:

$$Human \sqsubseteq \exists hasParents.Human$$

which is interpreted as, humans have human parents. Moreover, $\mathcal{SHIQ}$ allows for inverse roles, transitive roles and subroles. For example, the role $hasChild$ is inverse of the rule $hasParent$, the role $hasAncestor$ is transitive, and $hasParent$ is subrole of $hasAncestor$.

We can conclude this section by mentioning that, DL has a lot of advantages. Among them, is that it has a well-defined syntax, it is very expressive, it has well-defined semantics, and many DLs support decidable reasoning. Moreover, DLs allow one to represent incomplete knowledge, which is the type of knowledge we mostly deal with when we model the real world. It is also worth mentioning that, DLs model the domain in terms of concepts and relationships, which is very intuitive for representing ontology. These features make DLs a perfect choice for representing ontology. And this explains why DL languages are used as basis for the Ontology Web Language: OWL (Allemang and Hendler 2011).

2.3 Ontology-Driven Semantic Data Integration in Open Environment

The author in (Y. D. Wang 2009) discussed several issues involved in data integration in open environment. The issues addressed in (Y. D. Wang 2009) are namely; heterogeneity, the autonomy, and the distribution. The work proposes architecture called Service-Oriented Semantic-Driven Architecture (SOSDA) which is a web-based multi-agent multi-tier architecture to deal with these issues. In order to deal with the issue of heterogeneity, the authors adopted ontology to be the source of semantics. The author argues that the ontology may only be agreed on in closed environment. That is why the authors described each party's specification of the domain as the ontological view. And since several parties can have different ontologies (Ontological view according to the author), semantic transformation techniques are used to map these ontological views to one another. The authors defined semantic transformation to be “finding a function or mapping that assigns the elements of the target ontological view vocabulary to the elements of the source ontological view vocabulary”. This definition is usually called ontology mapping in the data integration community.

In (Y. D. Wang 2009), an extensional reduction technique is adopted for the modeling of the data integration systems (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009). As will be explained later in this work, that extensional reduction model is appropriate for describing systems in which the set of existing entities is not allowed to change while the relations between them may change. However, the extensional reduction model does not adequately describe information systems or dynamic systems in which new entities are allowed to enter and/or leave the world. It also describes the domain in terms of extensional entities rather than concepts.

The author then provided a formal definition for “ontological commitment of a view” and “ontological view”. These definitions are more or less overloaded versions of Guarino’s definitions of Ontology and Ontological commitment (Guarino, Oberle, and Staab 2009). The reason is that, the author’s argument about the ontology is meant to be an agreed on specification is not accurate. Ontologies designed for the same domain, or even describing a shared conceptualization can be different depending on who is specifying the conceptualization. As such, the motivation behind the definition of “ontological view” and “ontological commitment to a view” are deemed unintuitive.

The author then defined the formal definitions to the following terms: Transformable, Partially Transformable, and Untransformable, to be used as a base for the ontology transformation. The author then provides a formal definition for Ontology transformation (Mapping). The author then provided a definition for semantic equivalence as follows:

The predicate symbol p_S is semantically equivalent to the predicate symbol p_T if and only if the operands of p_S and p_T are correspondingly equivalent. To the author, the semantic equivalence, as defined above, is sufficient to map predicate symbols to one another.

Even though the author in (Y. D. Wang 2009) addresses the problem of semantic transformation only, and not semantic integration, there is some critical point that should be addressed more adequately. The main issue is that the authors adopted an extensional reduction model, which is found to be inadequate for modeling in open environment. The work in (Y. D. Wang 2009) also employed an elementary method for ontology matching. Elementary ontology matching ignores a lot of important details about the structure of the ontology. This is because ontologies have taxonomical structure and elements of an ontology inherit the semantics from their parents. The authors referred to the ontology as “ontology of a language” and defined it as a tuple <L, K_S> in which L is a language and K_S is an ontological commitment. We argue that a language and an ontological commitment do not uniquely define an ontology. For the same language and ontological commitment you may able to define several ontologies each of them approximates the intended model differently. And this is why we have good ontologies and bad ontologies (Guarino, Oberle, and Staab 2009). When the authors defined equivalence between predicates, two predicates are considered to be equivalent if they share an equivalent set of parameters. Since two different predicate can have the same set of parameters and yet are not intensionally equivalent, this definition of equivalence is not appropriate for open environment.

Another framework that addresses the ontology-driven semantic integration in open environment is found in (Xue 2010). The author in (Xue 2010) identified three research issues, namely; the architecture of a semantic integration enabled environment, ontological view modeling and representation, and Semantic equivalence relationship discovery (Mapping). Again, the main focus of the work was around the matching aspect which the authors refer to as “Semantic Equivalence Relationship Discovery”. The author in (Xue 2010) did not capture the semantics properly; rather, the work relied on the textual and syntactical data in the relational database schemas. It is also worth mentioning that, as is the case in (Y. D. Wang 2009), the author adopted an elementary method for ontology matching which ignores a lot of details about ontologies, including the taxonomical structure. Moreover, database schemas are built with a closed-world assumption. This makes inappropriate to rely on when generating an ontology, especially if the purpose is to model for an open environment. With regards of the ontology modeling and representation issue, the author adopted the extensional reduction model (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009) which is based on the “possible world” approach. The author also adopted the definition of ontological view and ontological commitment to a view from (Y. D. Wang 2009). The authors used the schema as a source of semantic. This was done by extracting data driven ontologies (DDO) from the data. This method can help in case of the sources do not have explicit ontologies. However, counting on DDO only in open environment means the schemas are the only source of semantics. Counting on schemas only as sources for semantics can result in inaccurate semantics. It is also worth mentioning that the implicit semantic in the database schemas cannot be maintained. The author defined the conceptualization to be “an abstract model that consists of the relevant concepts and the relationships that exist in a certain domain”. While this definition captures the intensional nature of a conceptualization, the use of the word model can imply the use of formal language, or the lead to the illusion of being something physical. The author also defined a Concept to be “anything that objectively exists in the real world and is rationally identified as existing in a conceptualization in terms of a domain of discourse”. To serve the matching between concepts of different ontologies, the author informally defined the equivalence between two concepts as follows: “Two concepts are semantically equivalent if their properties are the same or largely overlap”. The author also assumes that all the vocabularies used by the information models are based on natural languages. In turns, the author uses this in order to syntactically match different element of various schemas. The authors also discussed the extrinsic and intrinsic concepts. The extensional reduction model that the author adopted, however, does not describe the properties. And as such, it does not utilize the extrinsic concepts. When it comes to the representation of ontology, the author in (Xue 2010) choose a frame-based language to represent ontology (Xue, Ghenniwa, and Shen 2010). As has been shown earlier, frame language are limited in terms of their expressiveness and reasoning. As such, Description Logic has been long used to represent ontology in the knowledge engineering community (De Giacomo et al. 2018) and (Calvanese et al. 2018). Not only this, but also the author in (Xue 2010) also used pure XML to implement the frame-representation of ontology. By doing so, the author represents the ontology as a taxonomical structure with no explicit rules. The set of explicit rules is a very important part of the ontology. This is because it describes the relationships between the world concepts and supports reasoning and inference.

2.4 Ontology Mapping

Even though ontologies assign meanings to data and data structures, ontologies, created for the same domain, can still have certain degree of heterogeneity. This is because they can be built and maintained by different people. Mapping various ontologies to each other can help bridge this gab. This is done by finding matching between semantically related entities in different ontologies. Most of the approaches for ontology mapping and matching use elementary level techniques. In these techniques, each element of the ontology is separated and treated as a single entity out of the context. Usually these approaches do not result in promising results. The reason is that, dealing with different elements in isolation ignores a lot of details that can be utilized if the elements are in the correct context. These details are mostly related to the structure of the ontology. The structure of the ontology is very important to be considered while matching different ontologies. The reason is that, ontologies usually have taxonomical structure in which element inherit the semantics of their parents. In this type of structure, elements also pass their semantics to their children. In this review, we will discuss different ontology mapping techniques and try to decide which ones are preferred for the task of data integration in open environment.

(Choi, Song, and Han 2006) defined ontology merging, alignment, and integration as follows:

- *Ontology merging*: is the process of generating a single, coherent ontology from different ontologies related to the same subject.
- *Ontology alignment*: creates links between ontologies. Ontology alignment is made if the sources become consistent with each other but are kept separate. This is useful when the different ontologies have complementary domains.
- *Ontology integration*: Generating a single ontology in one subject from other ontologies in different subjects. The subjects of the different ontologies can be related.

Ontology mapping is a tool that is used to facilitate the process or merging, aligning, or integrating several ontologies. The authors then classified ontology mapping techniques into three different categories. These three categories are:

- *Ontology mapping between an integrated global ontology and local ontologies*: Maps concept found in one ontology into a view, or a query over other ontologies (usually, over the global ontology in LAV systems or over the local ontologies in GAV systems). In this class of algorithms, finding the mapping is usually a relatively easy task. This is because an integrated global ontology provides a shared vocabulary and all local ontologies are related to one global ontology.

However, this mapping can be hard to find among different ontologies which have mutually inconsistent information. This is because a global ontology cannot be created in that case.

- *Ontology mapping between local ontologies*: This category provides interoperability for highly dynamic, open, and distributed environments and is more appropriate for the Web. In this mapping ontologies keeps its content locally and it can provide interoperability between ontologies when they cannot be integrated or merged. Compared to mapping between an integrated ontology and local ontologies, this category mapping is more scalable. This is because the changes (adding, updating, or removing) of an ontology could be done locally without affecting other mappings. Finding mappings between local ontologies may not be easy because of the lack of common vocabularies. In open environment, when multiple ontologies cannot be merged because of mutual inconsistency of the information sources, this category of mapping can be used for them to interoperate.

- *Ontology mapping in ontology merge and alignment*: identifies similarities and conflicts between the various source (local) ontologies to be merged or aligned. This is considered the first step for ontology merging or alignment. After creating links between local ontologies while they remain separate, a single coherent merged ontology can be created through an ontology merging process. This mapping applies to ontologies over the same or overlapping domain.

It appears, from the above description of various types of ontology mapping algorithms, that, the third type, *ontology mapping in ontology merge and alignment*, is not appropriate for open environment. This is because it does require high degree of consistency and coherence between various ontologies. This is hard to find in open environment. On the other hand, the first type, *ontology mapping between an integrated ontology and local ontologies*, is more suitable when a mediated global ontology is generated from several ontologies. In open environment, this is not usually the case. In open environment however, there is no generation of common mediator global ontology.

However, if a collection of data sources in open environment will have a common mediator, this type of mapping can be employed to a portion of the network. The second type mentioned above, *ontology mapping between local ontologies*, however is more appropriate for a distributed open environment. This is because it does not make assumptions about the coherence or the similarities between ontologies. And since open environment has a loosely-coupled nature, it is likely that various ontologies possess certain degree of heterogeneity. As such this type of mapping is intuitively more appropriate for open environments. In what follows, various tools and techniques for ontology matching will be discussed in more details.

The authors in (Bouquet et al. 2003) distinguished between a context and an ontology and proposed an extension to the OWL language which will be able to explicitly describe mappings between various ontologies. The result is a C-OWL (Context OWL) in which ontologies are mapped using bridge rules. The authors in (Bouquet et al. 2003) proposed five bridge rules, these rules are:

- more general than ($\supseteq$)
- less general than ($\subseteq$)
- equivalent ($\equiv$),
- related or compatible ($*$)
- unrelated or incompatible ($\perp$)

The mapping in (Bouquet et al. 2003) is directional. Which means, M_{ij} from ontology O_i to ontology O_j is not necessarily the inversion of M_{ji} from ontology O_j to ontology O_i. According to the authors in (Bouquet et al. 2003), a mapping M_{ij} can be empty. An empty mapping represents the impossibility for O_j to interpret any concept from ontology O_i locally. The mapping M_{ij} might be a set of bridge rules of the form $i{:}x \overset{\equiv}{\rightarrow} j{:}y$ from an element *x* of ontology O_i to an element *y* of ontology O_j.

When *x* and *y* are concepts, say *C* and *D*, the intuitive reading of $i: C \overset{\equiv}{\rightarrow} j: D$ is that the *i*-local concept *C* is equivalent to the *j*-concept *D*. Whereas, the assertion $i: C \overset{\subseteq}{\rightarrow} j: D$ is intuitively read as, i-local concept *C* is more specific that j-concept *D*.

In (Bouquet, Serafini, and Zanobini 2003) the authors proposed a technique called ***CTXMATCH*** for the purpose of discovering semantic mappings across hierarchical classifications (HCs) using logical deduction. This algorithm takes two input hierarchies (H, and H1), and for each pair of concepts k ∈ H , k1 ∈ H1, returns their semantic relation (⊇, ⊆, ≡, ∗, and ⊥). The authors propose the usage of three distinct level of semantic knowledge. These levels are:

Lexical Knowledge: knowledge about the words used in the labels. For example, the fact that the word 'image' can be used in the sense of a picture or in the sense of a personal façade, and the fact that different words may have the same sense, 'picture' and image' for example.

Domain Knowledge: Knowledge about the relations between the senses of labels in the real world or in specific domain. For example, the fact that Tuscany is part of Italy or the fact that Florence is in Italy.

Structural knowledge: knowledge about how labels are arranged in a hierarchy. For example, the fact that certain concept is the part or child of another concept in the hierarchy.

As such, In CTXMTCH mappings can be assigned a clearly defined semantics and all the structural, lexical, and domain knowledge are considered in the discovery of the mapping between the concepts of two ontologies.

In (Silva and Rocha 2003), a technique called ***MAFRA*** (Ontology MAapping FRAmework for distributed ontologies in the Semantic Web) is proposed. The technique provides a distributed mapping process that consists of five horizontal and four vertical Modules. The horizontal modules are as follows:

- Lift & Normalization: to deals with language and lexical heterogeneity between source and target ontology.

- Similarity Discovery: establish similarities between entities of different ontologies.

- Semantic Bridging: It defines mapping for transforming source instances into the most similar target instances.

- Execution: It transforms instances from the source ontology into target ontology according to the semantic bridges.

- Post-processing: It takes the result of the execution module to check and improve the quality of the transformation results.

Whereas, the vertical modules are:

- Evolution: It maintains semantic bridges in synchrony with the changes in the source and target ontologies.

- Cooperative Consensus Building: It is responsible for establishing a consensus on semantic bridges between two parties in the mapping process.

- Domain Constraints and Background Knowledge: It improves similarity measure and semantic bridge by using WordNet or domain-specific thesauri.

- Graphical User Interface (GUI): Human intervention for better mapping.

MAFRA maps between entities in two different ontologies using a semantic bridge, which consists of concept and property bridges. The concept bridge translates source instances into target ones. The property bridge transforms source instance properties into target instance properties.

OKMS (Ontology-based knowledge management system) is another ontology matching technique proposed in (Maedche et al. 2003). The OKMS framework performs five-step to identify ontology-mapping. These steps are:

- Lift and normalization: If source information is not ontology-based, it will be transformed to the ontology level by a wrapper.
- Similarity extraction: The similarity extraction phase creates a similarity matrix, which represents the similarities between concepts and instances in ontologies being mapped.
- Semantic mapping: finds the mappings rules (How to transform one ontology to the other).
- Execution: Execute the mappings.
- Post-processing: attempts to improve the results of the execution phase.

The authors in (Besana, Robertson, and Rovatsos 2005) introduced a technique for P2P ontology mapping. The proposed technique facilitates interaction between various agents using mappings defined only for the portion of ontologies relevant to the interaction.

Another structural based technique for ontology mapping is the S-Match technique (Giunchiglia, Yatskevich, and Shvaiko 2007) and (*S-Match - Semantic Matching* 2014). In (Giunchiglia, Yatskevich, and Shvaiko 2007) the authors are focusing on semantic matching approach the search for semantic correspondences by using meanings (Concepts) rather than labels. This is what was mentioned in the previous paragraph. The reason is; if an element in the schema or the ontology is been mapped in isolation to other elements connected to it, most of the semantics that this element possess are ignored. This is because the context (location in the graph or taxonomy) does add a lot of semantics to every element in the hierarchy. This work also uses semantic similarity relations between concepts instead of syntactic similarity relations. So, this work considers relations, which relate the extensions of the concepts under consideration (for instance, more/less general relations). The authors provided reconstruction of some of the main matching problems and rearticulated them in terms of the generic problem of matching graphs. They also identified semantic mapping as an approach for generic matching; and proposed a decider for propositional satisfiability (SAT) as a method for implementing semantic matching. However, their proposed solution works only on

Directed Acyclic Graphs (DAG's) and links of type is-a. And that means, if the graph is more general and have cycles, this solution will not guarantee conversion. Also, this algorithm is not general to other types between nodes; i.e. has-a, etc.

Instead of just matching elements of ontology to other elements of a different ontology, the authors also try to find more complex mappings. In these mappings the authors try to discover some relations between elements in different ontologies such as whether or not an element of some ontology is a generalization or specialization of an element of a different ontology. The main problem is decomposed into two sub problems. These problems are:

- Extracting graphs from the data or conceptual models.
- Matching the resulting graphs.

The matching is then classified into syntactic and semantic matching. The key intuition with syntactic matching is to map labels of nodes and look for similarities using syntax driven technologies. As for the semantic matching, the meanings of the nodes are mapped. And the matching is done based on the concepts rather than labels.

When it comes to implementation; the authors have described two different level of granularity in implementing their work. The first is element-level, which means that each matching is done based on each element of the ontology isolated from the rest of it. The authors took advantage of the existing matching algorithms while using semantics instead of syntax for the purpose of semantic matching.

The second level of granularity is the structure-based matching. In this matching technique, the matching is done based on the context in which each concept lies within the ontology. The authors specified six steps for their matching algorithm. Assuming that we are working on two ontologies to be matched, these steps are:

1- Extract the two graphs

2- Compute element-level semantic matching

3- Compute the concepts at nodes

4- Construct the propositional formula

5- Run SAT

6- Iterations

Then it can be inferred that C1 = C2. An implementation of this algorithm is available online (*S-Match - Semantic Matching* 2014).

The Ontology Alignment Evaluation Initiative (OAEI) has been testing and evaluating the state of the art ontology matching techniques. We will discuss the major participating ontology matching systems, and the reader is referred to (Grau et al. 2013) for the results and comparison between different methods. Below, some of the high ranked ontology matching techniques will be described.

One of the ontology matching systems that have received very high rating in the OAEI2013 (Grau et al. 2013) is *YAM++* (Ngo and Bellahsene 2012) and (Ngo and Bellahsene 2013). YAM++ matcher tries to make use of all the useful information in the ontologies. This includes the terminological, structural (taxonomical or contextual), semantics, and extensional information. The main components of the YAM++ matcher are displayed in Figure 9.

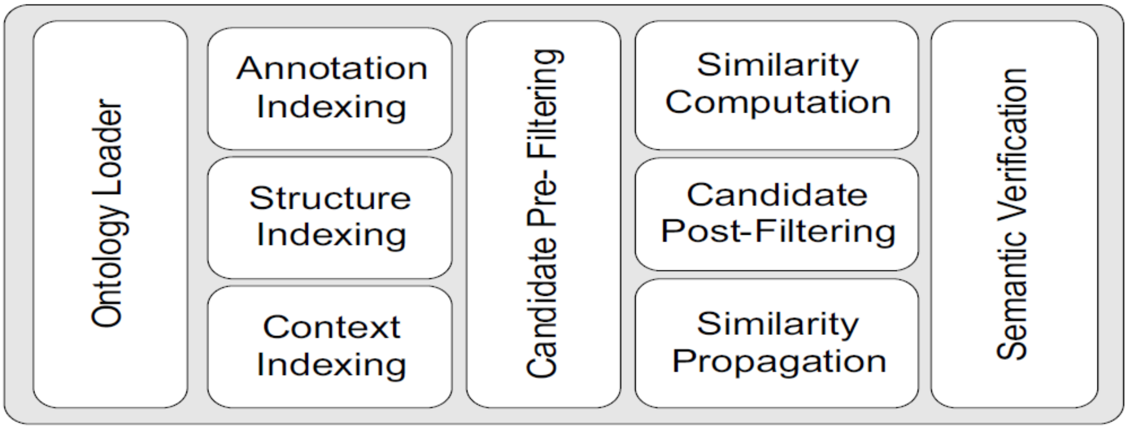

Figure 9: The main components of the YAM++ matcher

The workflow for a typical YAM++ ontology matching is described below:

1- Input ontologies are loaded and parsed by the Ontology Loader component. The ontology loader component uses an OWL 2 reasoner in order to discover hidden relations between entities in ontologies.

2- Various aspects of the information of the entities in the matched ontologies are indexed. The indexing includes:

   a. Annotation indexing, which extracts all annotation information of entities. Even with annotation information described in various natural languages, the Annotation indexing component still accounts for that.

   b. Structure indexing, which sorts the main structure information of ontologies such as IS-A and PART-OF.

   c. Context indexing, describes the entity, its parents, and its descendants.

3- In the initial screening step, the possible pairs of elements that are highly similar are filtered out by the Candidates Pre-Filtering component.

4- The candidate mappings are then passed into Similarity Computation component. The Similarity Computation component includes the following sub components:

   a. Terminological Matcher component that produces a set of mappings by comparing the annotations of entities

   b. The Instance-based Matcher component that supplements new mappings through shared instances between ontologies

   c. The Contextual Matcher, which is used to compute the similarity value of a pair of entities by comparing their context profiles.

   The matching results of the three subcomponents mentioned above are combined to have a unique set of mappings. Those are called, the element level matching results.

5- The element level matching results are used as input to the Similarity Propagation component. The Similarity Propagation component makes use of the structural information of entities. The resulting output of the Similarity Propagation component is called the structure level matching result.

6- The Candidate Post-Filtering component is then used to select the potential candidate mappings from element and structure level results.

7- Finally, the Semantic Verification component refines those mappings in order to eliminate the ones that are inconsistent.

Another ontology matching technique that received high ranking is the AML or Agreement Maker Light (Faria, Pesquita, Santos, Palmonari, et al. 2013) and (Faria, Pesquita, Santos, Cruz, et al. 2013). The AML matcher is an elementary-level ontology matching framework. It is a lightweight approach that is based on the AgreementMaker approach (Cruz, Antonelli, and Stroe 2009) and (Cruz et al. 2011). This framework is taking advantage of the fact that in many cases, the ontology matching task does not require a complicated structural-based approach. So, the AML approach tries to efficiently handle the task of matching large size ontologies efficiently. The current state of the framework does not include components for instance matching or translation. And as such, it cannot handle all ontology matching tasks. The AML framework matching lifecycle can described in terms of six main steps as shown in Figure 10.

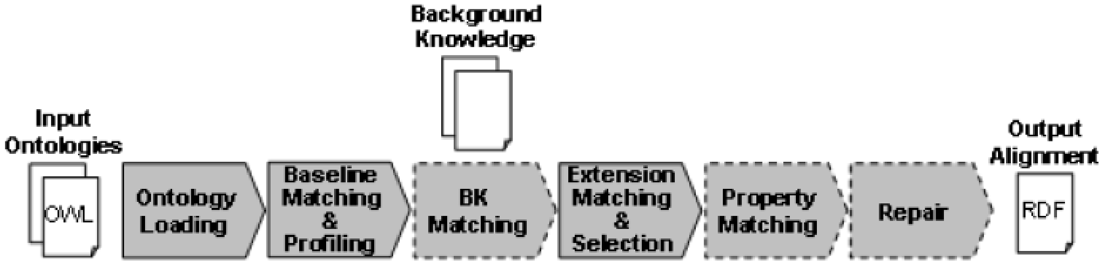

Figure 10: The Agreement Maker Light Workflow (Faria, Pesquita, Santos, Cruz, et al. 2013)

These steps are namely: ontology loading, baseline matching and profiling, background knowledge matching, extension matching and selection, property matching (conditional), and finally the repair step. Some of these steps are necessary and others are optional. This will be described in more details below.

1- Ontology loading: This step serves as a preparatory step for the next series of steps. In this step, the ontologies are read and the necessary information about each of the input ontologies is stored. This step reads the local Name, labels and synonym properties of all classes, normalizes them, and enters them into the Lexicon of that ontology. Then, it derives new synonyms for each name in the Lexicon by removing leading and trailing stop words and by removing name sections within parenthesis. After class names, AML reads the class-subclass relationships and the disjoint clauses and stores them in the RelationshipMap. Finally, AML reads the name, type, domain, and range of each property and stores them in the PropertyList. Note that AML currently does not store or use comments, definitions, or instances.

2- Baseline Matching and Profiling: In the baseline matching and profiling step, AML employs an efficient weighted string-equivalence algorithm, the Lexical Matcher, to obtain a baseline class alignment between the input ontologies. Then, AML profiles the matching problem by assessing the size (i.e., number of classes) of the input ontologies, the cardinality of the baseline alignment, and the property/class ratio. Regarding size, AML divides matching problems into three size categories (small, medium or large), which will affect decisions and thresholds during the background knowledge matching and the extension matching and selection steps. Regarding cardinality, AML also considers three categories (near-one, medium and high), which will determine how selection is performed during the extension matching and selection step.

   As for the property/class ratio, it determines whether AML will match properties during the property matching step.

3- Background Knowledge Matching: AML employs three sources of background knowledge: Uberon, UMLS, and WordNet. When using background knowledge, AML tests how well each source fits the matching problem by comparing the coverage of its alignment with the coverage of the baseline alignment. The Uberon Matcher uses the Uberon ontology (in OWL) and a table of pre-processed Uberon cross-references (in a text file). Each input ontology is matched both against the Uberon ontology using the Lexical Matcher and directly against the cross-reference table, and AML determines which form of matching is best (giving priority to the cross references, since they are more reliable). When Uberon is a good fit for the matching problem, it is selected as the only source of background knowledge and is used to extend the Lexicons of the input ontologies. When it is a reasonable fit, its alignment is merged with the baseline alignment.

   The UMLS Matcher uses a pre-processed version of the MRCONSO table from the UMLS Metathesaurus (in a text file). Each input ontology is matched against the whole UMLS table. Then AML decides whether to use a single UMLS source (by comparing the coverage of all sources) or the whole table. When UMLS is a good fit for the matching problem, its alignment is used exclusively, and the extension matching and selection step is skipped. Otherwise, if it is a reasonable fit, its alignment is merged with the baseline alignment.

   The WordNet Matcher queries the WordNet database for synonyms of each name in the Lexicons of the input ontologies, using the Jaws API. These synonyms are used to create temporary extended Lexicons, which are matched with the Lexical Matcher. Because WordNet is prone to induce errors, AML uses it only to extend the baseline alignment, meaning that it matches only previously unmatched classes.

4- Extension Matching and Selection: The extension matching and selection step comprises two matching sub-steps that alternate with two selection sub-steps. First, AML employs a word-based similarity algorithm, the Word Matcher, to extend the current alignment globally. This is followed by a selection algorithm to reduce the alignment to the desired cardinality. Then AML employs the Parametric String Matcher, which implements the Isub string similarity metric, to extend the resulting alignment locally (i.e., by matching the children, parents and siblings of already matched class pairs). This is followed by a final selection sub-step.

When the matching problem is profiled as ’large’, the Word Matcher is skipped because it is too memory intensive to be used globally, and its local use is subsumed by that of the Parametric String Matcher.

In the interactive matching track, AML employs an interactive selection algorithm, which asks the user for feedback about mappings in case of conflict or below a given similarity threshold, until a given number of negative answers is reached.

5- Property Matching: In the property matching step, AML matches the ontology properties. AML compares the properties’ types, domains and ranges, looking for mappings in the class alignment when the domains/ranges are classes. Then, if the properties have attributes in common, AML measures the word-based similarity between their names (as per the Word Matcher). Also; also WordNet when background knowledge is turned on.

6- Repair: In the repair step, AML employs a heuristic repair algorithm to ensure that the final alignment is coherent with regard to disjoint clauses. The repair algorithm was used by default in all OAEI tracks, except for the Large Biomedical Ontologies track where AML was run both with and without repair

Based on the discussion above, the S-Match ontology mapping technique (Giunchiglia, Yatskevich, and Shvaiko 2007) and (*S-Match - Semantic Matching* 2014) appears to be the most adequate technique to generate mappings between various ontologies in open environment. One of the reasons is that, the S-Match algorithm relies on the structure of the ontology while trying to find the ontology matching. Another important reason is that, the S-Match technique attempts to generate mappings that are more than just equivalence relations. The S-Match technique tries to find more complex relations between the various elements of ontologies. As an example, whether an element of an ontology is a generalization or specialization of another element. This is a very interesting feature to utilize in open environment where a high level of heterogeneity is anticipated. This is, in particular, a great advantage when the heterogeneity is due to different granularity levels in designing the ontology. Some ontologies can specify the concepts; *Thing*, *Living*, *Human*, *Employee*, *Professor*. On the other hand, other ontologies can specify only the concepts; *Human*, *Professor*, *Staff*. It will be a real advantage if the concept *Employee* is found, by the matching algorithm, to be a specialization of the concept *Human*.

Chapter 3

3 Formal Intensional Model for Conceptualization and Ontology

In this chapter, the issue of formal treatment of conceptualization and ontology is discussed in details. It is known that formal accounts of conceptualization and ontologies are essential and fundamental for knowledge representation and information engineering. In the past, this issue has been addressed by many researchers. The approaches that are proposed in the past are based on extensional logic or extensional reduction model. This chapter, however, highlights several limitations of their applicability for modeling conceptualizations in dynamic and open environments. This is due to several strong assumptions that are not adequate for dynamic and open environments. Intensional logic is found to be a natural and adequate choice for modeling in open environment. After pointing out the limitations of the extensional and the extensional reduction models, this chapter presents an intensional model for conceptualization. The proposed model is based on the theory of Properties Relations and Propositions (Bealer 1979). As opposed to being reduced to extensional functions, the proposed description takes the concepts, relations, and properties as primitives and, as such, irreducible. The proposed description is then extended to describe the world in more details by capturing the properties of the domain concepts. Based on the intensional account of conceptualization, formal intensional definitions for ontological commitment and ontology are also presented.

3.1 Different accounts of Conceptualization

A conceptualization is defined as “an abstract model that consists of the relevant concepts and the conceptual relations that exist in a certain domain (Xue 2010). This definition emphasizes the intensional nature of a conceptualization. Other definitions are also proposed by other researchers that reflect different accounts of conceptualization. In (Genesereth and Nilsson 2012), conceptualization is defined as "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relations that hold amongst them". This definition is also chosen by Gruber (Gruber 1993). This definition reflects an extensional account of a conceptualization. In (Genesereth and

Nilsson 2012) a formal description for conceptualization is presented. The description proposed in (Genesereth and Nilsson 2012) describes the conceptualization in terms of the objects in the world of interest and the extensional relations that do exist between them.

In (Guarino, Oberle, and Staab 2009) it is argued that a conceptualization is about concepts. And as such, the conceptualization should not change unless the meanings do change (Guarino and Giaretta 1995). In (Guarino and Giaretta 1995), a conceptualization is defined as "an intensional semantic structure that encodes the implicit roles constraining the structure of a piece of reality". This definition also shows that conceptualization is of an intensional nature. This is evident since the conceptualization is defined as a semantic structure. The keyword here is “semantic” which is concerned with meanings as opposed to the concrete objects. As will be shown later, intensional matters should be described with a logic that is compatible with its nature. And so, extensional logic cannot describe intensional contexts. And this is why an intensional notation is required for the task of describing a conceptualization.

An extensional reduction notation for describing a conceptualization is adopted in (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009). This extensional reduction model follows the possible world approach for intensional logic (Anderson 1984). The extensional reduction model is more adequate than the extensional model as it deals with conceptual relations as opposed to extensional relations in the extensional model. There are, however, several formal and intuitive concerns about the possible world approach that reduces the intensional entities to extensional ones as shown in (Bealer 1993) and (Bealer 1998b). It is also noticed that extensional reduction model is appropriate for describing systems in which the set of existing entities is not allowed to change. The advantage of the extensional reduction model over the extensional model is that it accounts for changes in the relations between the existing entities. However, the extensional reduction model does not adequately describe information systems or dynamic systems in which new entities are allowed to enter and/or leave the world. It also describes the domain in terms of extensional entities rather than concepts. The extensional reduction notation also treats the concepts as relations, which is found to be inappropriate and unintuitive. For these reason the need for an intensional-based notation for describing a conceptualization arises.

In this chapter, two different approaches for describing a conceptualization are discussed and analyzed. These approaches are fundamentally different as they belong to different classes of logic. The PRP theory (Bealer 1979) for intensional logic is then discussed. An intensional model for the conceptualization, based on the PRP theory, is proposed. This intensional model avoids the limitations of the extensional and the extensional reduction notations. The proposed notation is also extended to support a more fine-grained description of a conceptualization in which the properties of the domain concepts are captured. We will start with some important definitions that will pave the road for the rest of the chapter.

3.2 Definitions

Before we dive deep into the technical details of modeling the conceptualization, it is very important to define the main terms relevant to a conceptualization. Following are these definitions:

Conceptualization: “an abstraction that consists of the relevant concepts and the conceptual relations that exist in a certain domain”.

As mentioned earlier, the conceptualization is defined as abstract model that consists of the relevant concepts and relations that exist in certain domain (Xue 2010). This definition will be revised as “an abstraction that consists of the relevant concepts and relations that exist in certain domain”. We purposely remove the word “model” from the definition because it might imply the use of formal language, or the lead to the illusion of being something physical.

Concept: “Cognitive scientists generally agree that a concept is a mental representation that picks out a set of entities, or a category. That is, concepts refer, and what they refer to are categories” (Medin and Rips 2005). In other words, the term concept denotes a general, abstract, idea of a category inferred from the observation of its instances.

Particular: A particular, is a concrete entity that exists in space and time as opposed to a concept. When it is said that a name expresses its sense and designates its reference (Frege 1980), it should be understood that a concept corresponds to a sense, while a particular, or an instance, corresponds to the reference designated by this name. This does not mean that every instance of a category is exactly the same. But, only that from some perspective they are treated equivalently based on something they have in common.

Abstraction: The relation between a concept and particular will be referred to as abstraction. So a concept is created by keeping the characteristics that are common between several particulars while abstracting away the characteristics that are uncommon.

3.3 Extensional Logic vs. Intensional Logic

This section explains, briefly, the difference between extensional logic and intensional logic as applied to modeling a conceptualization. This is useful for clarifying the logic behind each one of the two models discussed below. After shedding some light on the difference between extensional and intensional logic, intensional logic will be explained in more details. Let us start by a simple example (Fitting 2004) and (Fitting 2006):

Example: If someone tells you that the Morning star is the Evening star, this changes your knowledge. This is because, now you know that the Morning star and the Evening star are equal. However, even though the two signs ("Morning star" and "Evening star") designate the same object, they do not have the same meaning. In this sense, meanings are the intensions, and things they designate are the extensions. A context that cares only about extensions is called an extensional context. On the other hand, if the context cares about the meanings, it is an intensional context (Fitting 2004).

One of the major differences that help distinguishing between the Intensional and extensional contexts is the applicability of substitutivity (Bealer 1982). In other words, a context in which substitutivity does not apply can be recognized as an intensional context. However, for extensional contexts, the substitutivity of equivalents always holds.

The following argument (Bealer 1982) explains the failure of the principle of substitutivity in the intensional contexts.

x believes that everything runs.

Everything runs if and only if everything walks.

∴ x believes that everything walks

It is obvious that the above argument is intuitively invalid. This is because the substitutivity is used in an intensional context in which it does not apply. Sentences like; “It is known that…”, “It is believed that …”, “It is said that…”, “It is necessary that…” are typical intensional contexts (Fitting 2004). For a computer scientist, expressing the belief of an agent or the knowledge of an information system follows the same rule. That is why the belief of an agent and the knowledge of an information system are intensional matters.

Classical first-order logic is extensional by nature. And so, for this class of logic, substitution of equivalent expressions preserves truth. Also, intensions are irrelevant to such systems. When such systems are used to describe a conceptualization they assume that the world consists of a set of entities and a set of extensional relations that hold between these entities at some instant. This is applicable if the interest is to describe a certain snapshot of the world of interest. However, extensional logic fails to describe the intensional account of a conceptualization.

Intensional systems, however, are those in which intensional features can be represented (Fitting 2004) . These are the systems that cannot be described in extensional logic. In order to describe such systems, several theories for intensional entities were proposed. Some of these theories included some reduction and some others adopted a non-reductionist view. Those theories, which incorporated reduction, reduce the intensional entities to extensional entities (Bealer 1998a). An example of such category of theories is the possible world approach as shown in (Anderson 1984) and (Lewis 1986). When used for describing a conceptualization, the reductionist approaches assume that the world has fixed set of entities. As such, these approaches are applicable if one is interested in describing a static system with a fixed set of entities in which the relations between objects are allowed to change. These approaches, however, are not adequate for describing information systems or dynamic systems in which entities or agents can enter and/or leave the system at any time.

The non-reductionist approaches, however, take the intensional entities such as concepts, relations, and properties, at face-values, i.e. as real irreducible entities. An example to theories of this category is the theory of Properties, Relations, and Propositions (PRP) (Bealer 1979), (Bealer 1982), and (Bealer 1993). Modeling the conceptualization using this class of logic is more adequate for dynamic systems and open environment. It allows for the description of intensional contexts such as the belief and the knowledge. It also accounts for the changes in the world as long the concepts and the meanings do not change.

3.4 Extensional Model for Conceptualization

As mentioned earlier, the conceptualization has been defined before as "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relations that hold amongst them" (Gruber 1993) and (Gruber 1995). This definition is based on the extensional model of conceptualization (Genesereth and Nilsson 2012). The extensional model is based on the extensional logic explained above. And as such it describes the conceptualization in terms of declarative sentences and ordinary relations. According to this model, a conceptualization is formally defined as a triple:

$$\mathcal{C}_e = (D_e, F_e, R_e) \qquad (1)$$

Here the subscript e refers to the extensional model. The triple in equation (1) consisting of a universe of discourse, a functional basis set for that universe, a relational basis set. The universe of discourse D_e is a set of all entities, or what is called extensions, in the domain. A function maps an entity $e_i \in D_e$ to another entity $e_i \in D_e$ based on an interrelation between the two entities. The set of functions that are emphasized in the conceptualization is referred to as the functional basis set F_e. And finally, the relational basis set R_e is the set of all extensional relations that hold between the elements of D_e.

The following example (Genesereth and Nilsson 2012) explains the extensional model of conceptualization:

Consider the blocks world that has only one concept (block). And consider a specific instance of this world in which there are five blocks arranged as shown in Figure 11

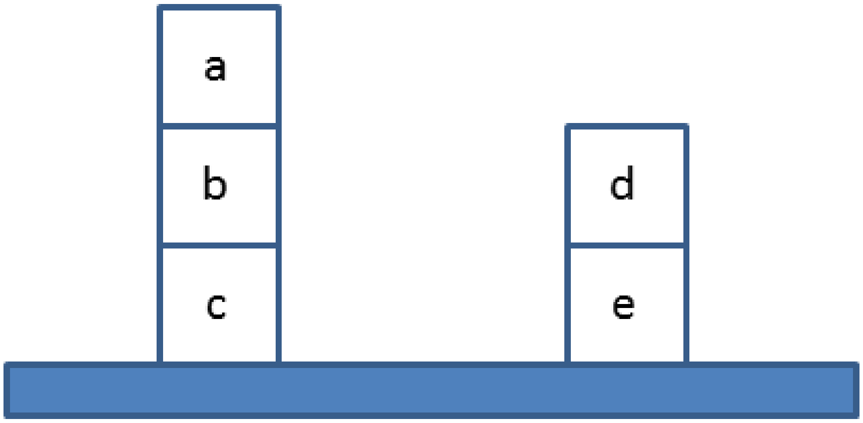

Figure 11: Five blocks on a table example (Genesereth and Nilsson 2012)

In this example, it is assumed that there is only one function in this domain that is relevant to the conceptualization. This function is called *Hat*, and it maps each item to its hat (the item that lies directly above it). It is also assumed that there are four different relations that are relevant to this conceptualization. These relations are *On*, *Above*, *Clear*, and *Table*. The conceptualization for this world, according to the extensional description, is:

$$\mathcal{C}_{e1} = (D_{e1}, F_{e1}, R_{e1}) \tag{2}$$

Where:

$$D_{e1} = \{a, b, c, d, e\} \tag{3}$$

And,

$$F_{e1} = \{hat_1\} \tag{4}$$

And,

$$R_{e1} = \{on_1, above_1, clear_1, table_1\} \tag{5}$$

The members of both (F_{e1} and R_{e1}) are ordered tuples on the elements of D_{e1}. In that sense:

- $on_1 = \{(a, b), (b, c), (d, e)\}$,
- $above_1 = \{(a, b), (a, c), (b, c), (d, e)\}$,
- $clear_1 = \{a, d\}$, and
- $table_1 = \{c, e\}$

In the previous example, the extensions, in the snapshot of the blocks world shown in Figure 11, were described using the extensional notation. It should be noticed, however, that the extensional logic cannot describe intensional matters. This is because extensional logic substitutes equivalent entities based on their extensions. And this does not apply to intensional contexts. For example, if we have a world that has two humans *John* and *Johnson*. This fact is represented extensionally as $Human = \{John, Johnson\}$ where both "$John$" and "$Johnson$" reference two humans. If both references "$John$" and "$Johnson$" happen to be doctors, then this fact will be described as $Doctor$= {"$John$", "$Johnson$"}. Because both "$Human$" and "$Doctor$" are mentioned in a declarative context, it is all about the reference and not the meaning. And as such, according to extensional logic; $Doctor = Human$. Even though the two signs "$Doctor$" and "$Human$" express two different meanings, they are treated as being equal. This shows how the extensional logic fails to describe the intensions as it focuses only on the extensions.

3.5 Extensional Reduction Model for Conceptualization

The fact that "an agent, or an information system, for simplicity, believes something about the world" cannot, adequately, be described using extensional logic. This is because it is an intensional context. And as such, describing such contexts using extensional logic might result in unintuitive arguments. (Guarino and Giaretta 1995) also pointed out that "the extensional notation of conceptualization is only useful if one is interested in an isolated snapshot of the world". For instance, if a different arrangement of blocks is considered, as shown in Figure 12, the corresponding conceptualization, according to the extensional notation, will be different.

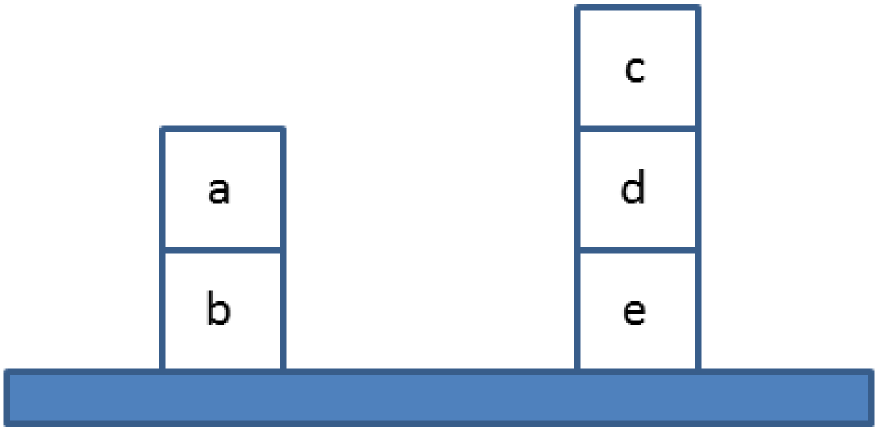

Figure 12: A different configuration for the five blocks (Guarino and Giaretta 1995)

It is argued in (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009) that, the conceptualization should focus on the meaning instead of a particular state of the world. And so, the conceptualization should not change when the arrangement of the blocks, in the blocks world, changes. This is because the meaning of the relation will still be the same regardless of the possible configurations of blocks.

In order to capture such intuition, extensional logic will not be sufficient. This is because the intensions do not matter for extensional logic. The extensional logic only cares about the designated objects or references and it does not care about the senses or intensions. For that reason, the possible world theory (Anderson 1984) is adopted as a basis for describing the conceptualization (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009). This theory reduces the intensional entities to extensional entities, i.e. extensional functions or sets (Bealer 1998a). The intuition behind the possible world reduction is that; in order to know the meaning of a sentence one would need to know the way things would have to be in order for the sentence to be true. A way things would have to be is referred to as a possible world (Anderson 1984).

(Guarino and Giaretta 1995) used the term conceptualization to denote “a semantic structure that reflects a particular conceptual view”. Using the possible world reduction, the conceptualization is then formally described in (Guarino, Oberle, and Staab 2009) as follows:

$$\mathcal{C}_{er} = (D_{er}, W_{er}, R_{er}) \tag{6}$$

We will use the subscript *er* to refer to the extensional reduction model. In this model, D_{er} is a domain of objects, W_{er} is a set of possible worlds, and R_{er} is a set of conceptual relations. According to this model, a conceptual relation of arity n on D_{er} is a function from the set of possible world W_{er} to the set $2^{D_{er}^{n}}$of all possible n-ary relations on D_{er}. It is also worth mentioning that, in this model the concepts are treated as relations, or functions, from W_{er} to $2^{D_{er}}$, where $2^{D_{er}}$ is the set of all unary relations on D_{er} (Guarino, Oberle, and Staab 2009).

Referring to the blocks world example shown in Figure 11, the conceptualization for the blocks world based on the extensional reduction model is described as follows:

$$\mathcal{C}_{er1} = (D_{er1}, W_{er1}, R_{er1}) \tag{7}$$

where:

$$D_{er1} = \{a, b, c, d, e\} \tag{8}$$

And,

$$W_{er1} = \{w_{11}, w_{12}, w_{13}, \dots\} \tag{9}$$

Is the set of possible worlds, i.e., the set of all possible configurations of the members of D_{er1}. And,

$$R_{er1} = \{block^{1}, clear^{1}, table^{1}, on^{2}, above^{2}\} \tag{10}$$

is the set of relations from $W_{er1}$to $\{2^{D_{er}}, 2^{D_{er}}, 2^{D_{er}},\ 2^{D_{er}^{2}}, 2^{D_{er}^{2}}\}$ respectively.

The superscripts in equation (10) specify the order of the relation. In that sense, unary relations have the superscript 1, and binary relations have the superscript 2, and so on. In order to show that the extensional reduction model has an advantage over the extensional model, the configuration in Figure 12 will be described according to the two models. The conceptualization for the world shown in Figure 12, according to the extensional model, is:

$$\mathcal{C}_{e2} = (D_{e2}, F_{e2}, R_{e2}) \tag{11}$$

Where:

$$D_{e2} = \{a, b, c, d, e\} \tag{12}$$

And,

$$F_{e2} = \{hat_{2}\} \tag{13}$$

And,

$$R_{e2} = \{on_{2}, above_{2}, clear_{2}, table_{2}\} \tag{14}$$

The members of the two sets, $F_{e2}$and R_{e2}, are ordered tuples on the elements of D_{e2}. In that sense,

- $on_{2} =\ \{(a, b), (c, d), (d, e)\}$,

- $above_2 = \{(a,b),(c,d),(c,e),(d,e)\}$,
- $clear_2 = \{a,c\}$, and
- $table_2 = \{b,e\}$

Here it is noticed that $D_{e1} = D_{e2}$, however, $R_{e1} \neq R_{e2}$, and in turns $\mathcal{C}_{e1} \neq \mathcal{C}_{e2}$. According to (Guarino and Giaretta 1995) “this is what originates the troubles”. This is because the conceptualization is about concepts and should not change if the state of the world changes (Guarino and Giaretta 1995). On the other hand, the configuration in Figure 12, described using the extensional reduction model, is:

$$\mathcal{C}_{er2} = (D_{er2}, W_{er2}, R_{er2}) \qquad (15)$$

And based on the possible world reduction, it can be shown that $D_{er1} = D_{er2}$. This is obvious since the entities in the world have not changed, i.e. the five blocks in both Figure 11 and Figure 12 Since W_{er} is the set of all possible configurations of the elements of D_{er}, and since $D_{er1} = D_{er2}$, it can also be shown that $W_{er1} = W_{er2}$. And finally since R_{er} is a set of relations from W_{er} to $2^{D_{er}^n}$. It is also obvious that, R_{er1} and R_{er2} are equivalent. And in turns, ($\mathcal{C}_{er1}$ and $\mathcal{C}_{er2}$) are the same conceptualization as one would expect.

3.6 Intensional Model for Conceptualization

It is clear that the extensional reduction, or the possible world approach, is more expressive as compared to the extensional model. As discussed in the previous section, different arrangements of the same entities will not result in different conceptualization. This is because the meaning of the relations between them does not change. However, for several reasons, this model needs to be further revisited, especially in the context of knowledge formalization, information systems, information integration, and open environments.

There are several formal and intuitive concerns about the possible world reduction (Bealer 1993) and (Bealer 1998a). First and foremost is that, it is a reduction that reduces the intensional entities to extensional entities. This reduction makes the propositions merely extensional functions from the possible world to truth values. It also treats the properties as extensional functions from individuals to propositions, and so on (Bealer 1993) and (Bealer 1998a). In (Bealer 1998b), it is also mentioned that "The possible world reduction also implies that a proposition cannot be simultaneously necessary and a posteriori. According to scientific essentialism however, that very proposition must be both necessary and a posteriori". Moreover, in (Bealer 1998a) it is also stated that "the possible world reduction fails for the properties of being necessary. And, in general, it fails for every iterable property". Further discussions about the possible world reduction can also be found in (Adams 1974) and (Jubien 1988) (as cited in (Bealer 1993)). Bealer proposed a non-reductionist formulation for intensional logic that is combatable with actualism as opposed to possibilism. The theory of Properties Relations and Propositions (Bealer 1979), (Bealer 1982), and (Bealer 1993) takes the properties, relations and propositions as real irreducible intensional entities.

Before the formal description is proposed, some important definitions will be discussed first. We will start with the definition of a concept as it is an essential element of a conceptualization. "Cognitive scientists generally agree that a concept is a mental representation that picks out a set of entities, or a category. That is, concepts refer, and what they refer to are categories" (Medin and Rips 2005). In other words, the term concept denotes a general, abstract, idea of a category inferred from the observation of its instances. A particular, is a concrete entity that exists in space and time as opposed to a concept. When it is said that a name expresses its sense and designates its reference (Frege 1980), it should be understood that a concept corresponds to a sense, while a particular, or an instance, corresponds to the reference designated by this name. This does not mean that every instance of a category is exactly the same. But, only that from some perspective they are treated equivalently based on something they have in common. The relation between a concept and particular will be referred to as abstraction. So a concept is created by keeping the characteristics that are common between several particulars while abstracting away the characteristics that are uncommon.

This relation between a concept and a particular is, in some sense, similar to the relation between an intension and extension. This later is called extensionalization. It is also, in a sense, similar to the relation between a class and an object; which is called instantiation.

A conceptualization is also defined as an abstract model that consists of the relevant concepts and the conceptual relations that exist in a certain domain (Xue 2010). Again this definition emphasizes the fact that the conceptualization is about concepts and meanings. And so, the conceptualization should remain the same even when the state of the world is changed or a particular is introduced to the world. This assumes that the new particular that is introduced to the system is abstracted to, or is an extension of, a concept that is already captured in the conceptualization. It is only the introduction of a new concept or conceptual relation that should result in different conceptualization. Having said that, even if a new particular is introduced to the world and this particular has a relation or a property that is irrelevant to the conceptualization, this relation or property should be abstracted out and the conceptualization will not be affected.

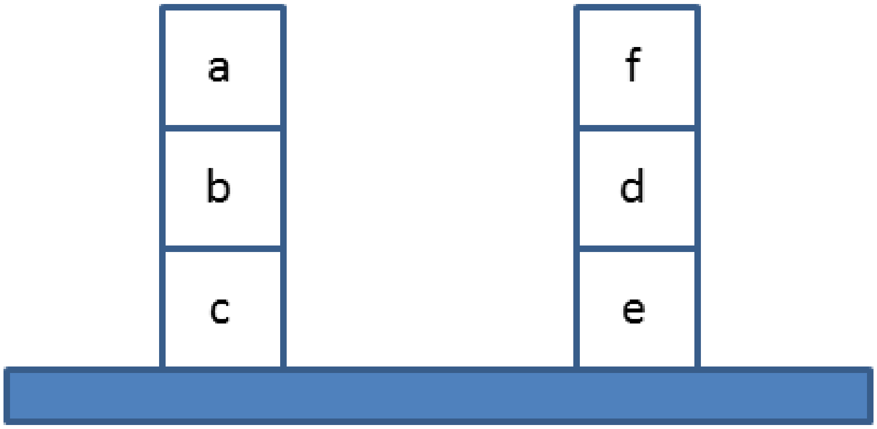

Figure 13: The blocks world with 6 entities instead of 5

In order for this point to be clear, Figure 13 shows another example of the blocks world in which another block f is introduced to the world. Let us assume that the conceptualization for the configuration shown in Figure 3, based on the extensional model, is $\mathcal{C}_{e3}$. As discussed before, it is quite evident that $\mathcal{C}_{e1}$, $\mathcal{C}_{e2}$, $\mathcal{C}_{e3}$ are different. This is partially taken care of in the extensional reduction model that is based on the possible world approach. The conceptualization for the configuration in Figure 13, based on the extensional reduction model, will be referred is described as follows:

$$\mathcal{C}_{er3} = (D_{er3}, W_{er3}, R_{er3}) \qquad \textbf{(}16\textbf{)}$$

Where, D_{er3} is defined as follows:

$$D_{er3} = \{a, b, c, d, e, f\} \qquad \textbf{(}17\textbf{)}$$

As mentioned earlier, $\mathcal{C}_{er1}$ and $\mathcal{C}_{er2}$ are equivalent. However, from (8), (12), and (17) it is clear that $D_{er1} = D_{er2} \neq D_{er3}$. And since W_{er} is defined as all possible configurations of elements of the domain D_{er}, then $W_{er1} = W_{er2} \neq W_{er3}$. Moreover, because R_{er} is a set of relations from W_{er} to $2^{D_{er}^{n}}$, it will be easy to show that $R_{er1} = R_{er2} \neq R_{er3}$ and in turns $\mathcal{C}_{er1} = \mathcal{C}_{er2} \neq \mathcal{C}_{er3}$.

Since the extensional reduction model is based on the possible world reduction, it is easy to show that the so called conceptual relations are, in fact, extensional relations between the set of possible world and the set of extensions in the domain. It is also clear that, the introduction of a new extension to the world changes the conceptualization. According to the intensional model, introducing a new particular, which corresponds to a concept that is already captured in the system, should not change the conceptualization. This is because the new particular in that case is merely another extension to the concept that is already known to the conceptualization.

Based on the above discussion, an intensional model that accounts for the instantiation, or extensionalization, is required. Being an intensional model, it should take the relations as intensional entities rather than reducing them to extensional functions. The intensional model should also capture the concepts (based on the observation of the particulars) instead of capturing the particulars themselves. Especially in the context of information systems, inserting a record in the database, for instance, can be considered as sort of instantiation. And this should not affect the conceptualization.

It is also worth mentioning that, the extensional reduction model treats the concepts as relations and mix them with the set of relations. We find this inappropriate and unintuitive for the purpose of this work. This is because, the concepts are abstractions of entities that exist in certain time and space while the relations are abstractions of the interrelations between these entities. Even though both concepts and relations are intensional entities, they are different in nature.

Another observation is that, the relations in the possible world approach are separated from the domain. And even though this is not wrong for describing the conceptualization, we adopt the view that the relations are intensional entities and should be taken as primitive, irreducible, entities (Bealer 1979) and (Bealer 1982) . And as such, it is more adequate to treat the intensional relations as part of the domain. In that sense, both the set of concepts and the set of conceptual relations will be members of the domain.

Finally, it is also important that the model can be expanded in order to describe the world in more details. An example of this would be a model that describes the properties of the concepts as will be shown later.

Motivated by the above observations, a new intensional model for describing the conceptualization is proposed. This model is based on Bealer’s intensional logic (Bealer 1979) and (Bealer 1982). The following section will shade some light on Bealer’s intensional logic. Then the proposed model will be explained. The proposed model will then be extended to describe the properties assigned to the concepts.

The following section will describe, in more details, the intensional logic, the intensional formal language, and the intensional semantics from (Bealer 1979) . This logic plays a very important rule in both the intensional model of conceptualization and the modeling of the data integration systems.

3.6.1 Intensional logic

Bealer’s formulation of properties relations and propositions (PRP) addresses not only logical modalities, but intensional matters as well. This section introduces the reader to Bealer’s intensional FOL as described in (Bealer 1979) and (Bealer 1982) . Let us start by two motivating examples from (Bealer 1979):

(1)

Whatever x believes y believes

x believes that A.

$\therefore$ ***y believes that A.***

(2)

Being a bachelor is the same as being an unmarried man

$\therefore$ ***It is necessary that all and only bachelors are unmarried men.***

Both of the above intuitively valid arguments are not expressible in standard first order predicate logic. However, they can be expressed in higher-order logics which are incomplete (Bealer 1979). Having a first order logic that is capable of expressing such arguments can avoid the troubles that arise from using higher-order logics. As discussed in (Bealer 1979), there are two conceptions of PRP. The first conception considers intensional entities to be identical if and only if they are necessarily equivalent. Since this conception does not impose any constraints on what is to count as a correct definition, both of the following are considered as correct definitions:

(a) ***x is grue if x is green if examined before t and blue otherwise.***

(b) ***x is green if x is grue if examined before t and bleen otherwise.***

According to the second conception an intensional entity when defined completely, must have a unique and non-circular definition. In that sense, looking at arguments (a) and (b) above, (a) alone is correct. Both (a) and (b) together are incorrect though. This is because they do not satisfy the non-circularity constraint. This conception is ideally suited for intensional matters (Bealer 1979). The author also proposes a new logic language by adding an intensional abstraction operator to the first order predicate logic. The result is a logic for PRP which is not only adequate for describing the logical modalities, but the intensional matters as well.

3.6.1.1 Formal Language

The formal language for the first order intensional logic L_w is defined in (Bealer 1979) as follows:

- Primitive symbols:
    - Logical Operators: &, $\neg$, $\exists$
    - Predicate letters: F_1^1, F_2^1 ... F_m^n.
    - Variables: x, y, z…
    - Punctuations: (,), [,]
- Definition of Term and Formula:
    - All variables are terms.
    - If t_i,., t_j are terms, then F_i^j (t_1, t_j) is a formula.
    - If A and B are formulas, then (A & B), $\neg A$, and ($\exists v_k$)A are formulas.
    - If A is a formula and vi,.., vm, $0 < m$, are distinct variables, then $[A]_{v,...vm}$ is a term.

From the definition above, it is clear that L_w is similar to the standard FOL except for the addition of the singular term $[A]_{v,...vm}$. A singular term $[A]_{v,...vm}$ denotes a proposition if $m = 0$, denotes a property if $m = 1$ and denotes an m-ary relation if $m > 1$.

Now, going back to the intuitive examples described above; they can be expressed in L_w as follows:

$$\frac{\begin{array}{l}(\forall z)\big(B(x,z) \supset B(y,z)\big)\\ B(x,[A])\end{array}}{\therefore B(y,[A])}$$

$$\frac{[B(x)]_x = [(U(x) \& M(x))]_x}{\therefore N([(\forall x)(B(x) \equiv (U(x) \& M(x)))])}$$

The use of the singular terms in the above two examples made it possible to express the arguments in first order logic. This avoids the complexity of higher order predicate logics. The author in (Bealer 1979) also developed a semantic method for the intensional logic language. What is interesting about this language is that it is able to describe intensional matters and is both sound and complete at the same time. We refer the reader to (Bealer 1979) and (Bealer 1982) for more details.

3.6.1.2 Intensional Semantics

As mentioned before, the syntax of the intensional first order logic is similar to that of the first order predicate logic except for the addition of the singular terms. As such, understanding the semantics of the singular terms is in the heart of understanding the semantics of the intensional logic language.

As is mentioned earlier, every formula in L_w is a formula in standard first order logic except for the singular term occurring in the formula. Therefore, the illustration here will explain only the semantics of the singular terms in L_w. So basically, characterizing the denotation of a singular term so that a singular term $[A]_{v1\ldots vn}$ will denote an appropriate proposition, property, or relation is what is going to be explained here. This will depend on the value of *m*. And because L_w can have infinite number of singular terms $[\mathrm{A}]_\alpha$, the specification for the denotation relation of L_w will be done recursively. In order to achieve this task, the singular terms are going to be arranged in order according to their syntactic complexity. For example, just as the complex formula $((\exists x)Fx \ \& \ (\exists y)Gy)$ is the conjunction of the simpler formulas $(\exists x)Fx$ and $(\exists y)Gy$, it will be said that, the complex term $[(\exists x)Fx \ \& \ (\exists y)Gy]$ is the conjunction of the simpler terms $[(\exists x)Fx]$ and $[(\exists y)Gy]$. In the same manner, just as the complex formula $-(\exists x)Fx$ is the negation of the simpler formula $(\exists x)Fx$, it will be said that, the complex term $[-(\exists x)Fx]$ is the negation of the simpler term $[(\exists x)Fx]$. The complex singular terms of L_w that are syntactically simpler than all other complex singular terms are those whose form is $[F_h^m(v_1, . . . , v_m)]_{v1, . . . , vm}$. These terms will be called elementary singular terms. The denotation of such an elementary complex term is just the property or relation expressed by the primitive predicate F_h^m. The denotation of a more complex term $[A]_\alpha$ is defined in terms of the denotations of the relevant syntactically simpler terms until an elementary term is reached.

3.6.2 Intensional Description for Conceptualization

The theory of PRP is a non-reductionist intensional formalization for intensional logic. This formalization takes the properties, the relations, and the propositions as real irreducible entities instead of reducing them to extensional entities. According to the theory of PRP, an intensional algebra is a structure (D, J, K) consisting of a domain D, a set of logical operations J, and a set of possible extensionalization functions K (Bealer 1979) and (Bealer 1998a). The domain D divides into subdomains that include the intensional entities of the domain. The set of logical operations includes, but not limited to, conjunction, negation, singular predication, existential generalization, and so on. And the possible extensionalization functions assign extensions to relevant items in the domain. Extensionalization can be defined as the process of keeping the abstraction distinct and maintaining the relationship between the abstractions and observed facts (Aparasu 2011). In other words, extensionalization is the connection between reality and the perception of the observer.

As mentioned earlier, the conceptualization is defined as abstract model that consists of the relevant concepts and relations that exist in certain domain (Xue 2010). This definition will be revised as “an abstraction that consists of the relevant concepts and relations that exist in certain domain". We purposely take off the word model from the definition because it might imply the use of formal language, or the lead to the illusion of being something physical. In order to intensionally describe conceptualization, an intensional structure, based on the theory of PRP, is used. This structure is formally explained below and various advantages of the new model are discussed.

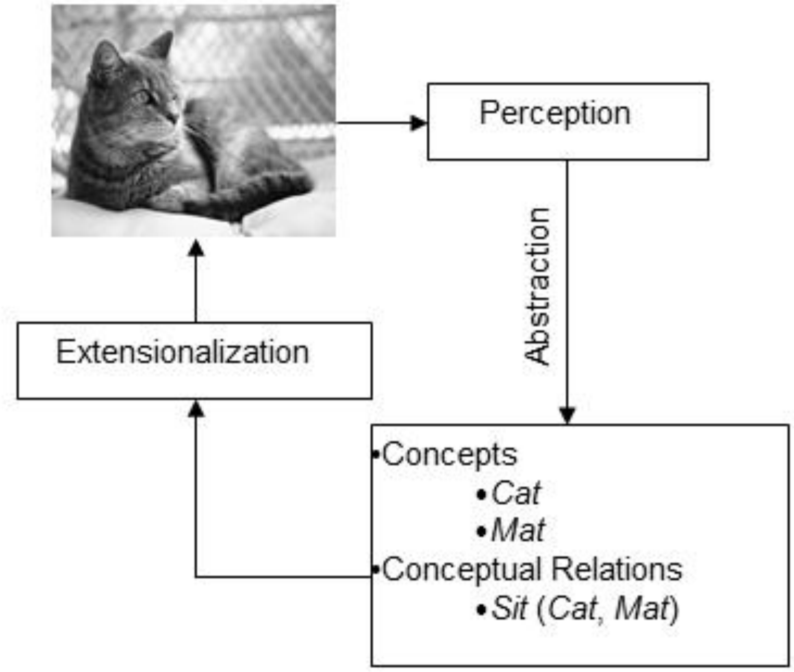

Figure 14: The relation between the conceptualization and the reality

According to the intensional notation, a conceptualization is described as follows:

$$\mathcal{C}_i = (D, K) \qquad \textbf{(}18\textbf{)}$$

in which D is a domain and K is a set of extensionalization functions. The domain D, in turns, consists of the set of concepts C and the set of conceptual relations R, written as:

$$D = (C, R) \tag{19}$$

The set of concepts *C* in (19) captures abstracts to all relevant entities in the world. And the set of relations *R* can be further decomposed into binary relations R_2, ternary relations R_3, and so on. The members of the set of extensionalization functions *K* assign entities of the reality to the corresponding concepts and conceptual relations in the conceptualization. Figure 14 explains how an extensionalization function relates elements of the reality to both concepts and intensional relations in the conceptualization.

Figure 14 shows how the particulars are related to the conceptualization through the extensionalization function. Note that, the predicate *Sit* (*Cat*, *Mat*) does not describe certain instances of the concepts *Cat* or *Mat*. Rather it intensionally means that entities corresponding to the concept *Cat* can be described as *Sit*ting on any entity that can be referred to as a *Mat*. And as such the conceptualization corresponding to the world in Figure 14 can be described as follows:

$$\mathcal{C}_4 = (D_4, K_4) \tag{20}$$

In that case, D_4 will be described as follows:

$$D_4 = (\{Cat, Mat\}, \{Sit\ (Cat, Mat)\}). \tag{21}$$

The question now is, what changes to reality should affect the conceptualization? Or in other words, when should the conceptualization change? In order to answer this question Figure 14 and Figure 15 are closely examined. In Figure 15, one can see two cats sitting on a mat. Is the conceptualization that describes the world in different from the one that describes the world in Figure 14? In order to answer this question we need to answer the following questions first:

Did the world change? If yes then:

a. Were extensions of new concepts introduced to the world? If yes, then:

   i. Are these concepts relevant to our conceptualization?

b. Were extensions of new relations introduced to the world? If yes, then:

i. Are these intensional relations relevant to our conceptualization?

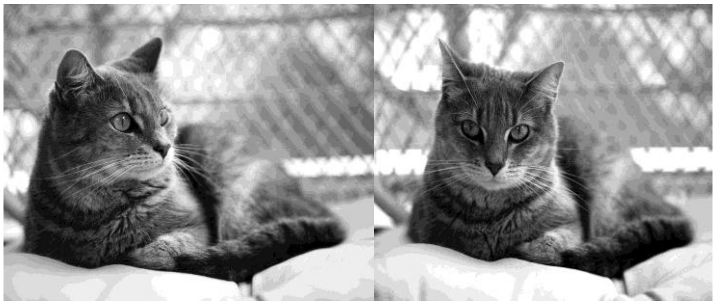

Figure 15: Two cats sitting on a mat.

By looking at Figure 15; the answer to the first question is YES. This is because another cat is now sitting on the mat. However, since the concept that is already captured in E_{i4}, this should not change the conceptualization. This is because the introduction of a new cat does not change the meaning of the concept cat. Now let us examine the relations between relevant concepts in Figure 15. There seem to be a relation between the two cats, as one of them is beside the other. Now, if this relation is relevant to our conceptualization, this will be perceived as a binary relation on the concept *Cat*, i.e. *SidebySide*(*Cat*, *Cat*). However, if this relation is irrelevant to our conceptualization, it will be abstracted out and the conceptualization $\mathcal{C}_{i4}$ will be able to describe the *Cat*/*Mat* world in Figure 15. And as such, our conceptualization captures the facts that, there can be cats, and there can be mats, and cats can set on mats. No matter how many cats, how many mats, and how many cats are sitting on mats, this should not affect the conceptualization.

By examining Figure 16 and trying to answer the same questions above, one can observe that the world has changed. This change adds both an extension of new concepts *Dog*, and extensions of new conceptual relations, i.e. *SidebySide* (*Dog, Cat*) and *Sit* (*Dog*, *Mat*). The next question would be, is the concept *Dog* relevant to our conceptualization? If the answer is No, then the concept *Dog* will be abstracted out and the conceptualization won't be affected. However, if the concept *Dog* is relevant to our conceptualization then the conceptualization should change in order to account for a new *concept*. In a similar way, we will need to answer the question about the conceptual relations and whether or not they are relevant to our conceptualization.

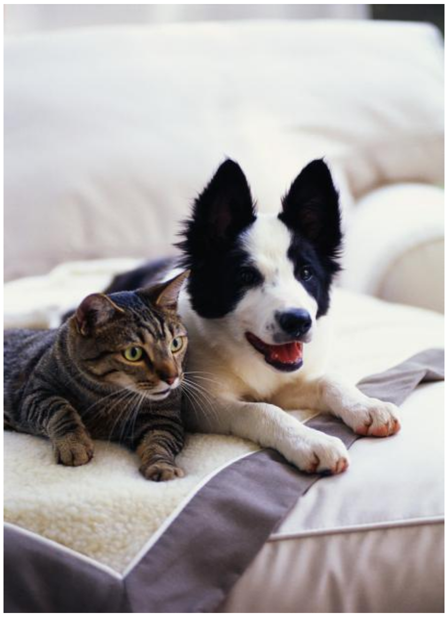

Figure 16: A cat and a dog sitting on a mat.

In a nutshell, the conceptualization is about concepts and meanings. It should not change unless two conditions satisfy. First, an extension of a new concept or an extension of a new conceptual relation is introduced to the world. Secondly, the new concept or the new conceptual relation is relevant to the conceptualization. However, if the change on the world does not change the concepts or the conceptual relation, this should not affect the conceptualization. Having said that, by revisiting the different configurations of the blocks world example shown in Figure 11, Figure 12, and Figure 13 one can observe the following:

1- The *extensional* model treats the three configurations as three different conceptualizations.

2- The *extensional reduction* model considers the first two configurations, shown in Figure 11 and Figure 12, to have the same conceptualization. However, the configuration shown in Figure 13 is considered to have a different conceptualization.

3- The *intensional* model considers the three configurations to have the same conceptualization. This is because this model descries the world in terms of *conceptual relations* and *concepts*. And since it is obvious that the concepts and conceptual relations captured in Figure 11, Figure 12, and Figure 13 are the same. The same conceptualization should be able to describe all of them.

3.6.3 Advantages of the Intensional Model

The extensional reduction model (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009) , based on the possible world approach, is more appropriate for describing the conceptualization as compared to the extensional model. This is because it accounts for the change in the relations between the entities in the world. The extensional reduction model, however, is suitable for describing static systems in which the configuration of the system may change, without introducing new entities. For the sake of describing information systems, or dynamic system that exist in open environment, the extensional reduction model may not be a good candidate. For this reason and for several reasons, mentioned above, the need for an intensional-based model is quite evident.

The intensional model has further improved the description of conceptualization so that it describes the relations as real irreducible entities instead of reducing them to extensional functions. It also deals with concepts rather than extensional entities or objects. Moreover, the intensional model separates the concepts from the relations as they are different in nature. This is different from the extensional reduction model which treats the concepts as relations. Furthermore, since the intensional model treats the intensional relations as primitive entities, they are considered to be part of the domain. It is also worth mentioning that the use of the singular term in the intensional logic (Bealer 1979) avoids higher order syntax for intensional logic (Majkić 2009). And finally, the proposed intensional description of conceptualization is easy to expand so that it describes more details about the world. In the next section, it will be shown how the intensional model will be expanded to describe the properties of the domain concepts.

3.6.4 Fine Grained Description

In (18), the intensional model of conceptualization describes the conceptualization as a tuple (*D*, *K*). In this description, *D* is composed of subdomains containing both the concepts *C*, and the conceptual relations *R*. This model can further be extended to descript not only the relations between concepts, but also the properties of the concepts themselves. The properties of a given extension are referred to as abstract particulars or tropes (Bacon 2008) and (Maurin 2013). Extensions of relations can also be referred to as tropes. And so both the properties and relations are considered subdomain of the domain *D*. This notation follows the PRP theory (Bealer 1979) and (Bealer 1998b) in which the *properties* are taken as primitive entities and considered as part of the domain. The values assigned to the properties can be thought of as concepts. However these concepts are different from the primary concepts in the domain. The difference is that these concepts are not of direct relevance to the conceptualization. And as such, they will be treated differently. The values that are associated to the properties are going to be called extrinsic concepts *Ce*. On the other hands, the concepts that are of direct relevance to the domain will be referred to as intrinsic concepts C_i.

The expanded model the domain will expand to capture the properties of the concepts. The domain *D* in the fine grained description will be described as:

$$D = (C_i, C_e, R, P) \qquad \textbf{(22)}$$

These four members in (22) represent *intrinsic concepts*, *extrinsic concepts*, *relations*, and *properties*. An example to a property in Figure 14 would be the color of the cat.

Assuming that a *Cat* can have one of several colors (*Black*, *White*, *Grey*, or *Brown*), these colors are considered extrinsic concepts in our conceptualization. The fact that cats can have the grey color will be described by the property *Color* (*Cat*, *Grey*). This should not be confused with asserting certain fact about a certain entity in the world. However, this should be understood as a conceptual relation that can be read as "extensions of the concept *Cat* can be attributed as having a *Grey Color*".

As shown in this section, the granularity of the intensional notation of the conceptualization can easily be controlled. This is an amazing property that allows scalability and gives more control on the description of a system. This attribute is used here to describe the properties of the domain concepts. However, we expect this property to offer flexibility in describing even more details about the system. Any details that can be described as part of the conceptualization need to be about the concepts and the meanings rather than the extensions. And, any extensions, if they are relevant to the conceptualization, will be abstracted and related to a member of the conceptualization, i.e. a concept, a relation, or a property.

3.6.5 Ontological Commitment and Ontology

This section provides an intensional account of ontological commitment and ontology. Since an ontology specifies a conceptualization, it commits to that conceptualization through an ontological commitment. As defined in (Rayo 2007) an ontological commitment for a sentence is what a sentence requires in order to be true. In the context of the reference theory; an ontological commitment is defined as a relation that holds between a sentence and an object (Parsons 1967) . In that sense, this object needs to exist in order for the sentence to be true. And so, it can be said that, this sentence is committed to that object.

Ontological commitment is also defined in (Parsons 1967) as a relation that holds between a sentence and a class of objects. These two definitions are based on two different accounts of ontological commitment. The first account is the extensional account (Jubien 1974). According to this account of ontological commitment, a language commits to extensions or objects in the domain of interest. The second account of ontological commitment is the intensional account (Jubien 1975). According to the intensional account, a language commits to kinds instead of particulars. In general, it can be said that an ontological commitment is a relation that holds between a language and intension. This intension can be a concept, a property, or an intensional relation. It is also shown in (Jubien 1972) , that ontological commitment is either intensional or inadequate.

Before the intensional account of ontological commitment is presented, let us first examine an example to the extensional account. In (Guarino, Oberle, and Staab 2009) , ontological commitment is defined as a structure $K = (C, I)$. In that structure, the interpretation function I is a total function $I : V \rightarrow D \cup R$. this interpretation function maps each vocabulary symbol to either an element of D or a relation belonging to the set R. As discussed in sections 3.6.2 and 3.6.4, the elements of D are extensions and not concepts. And so, this structure commits to extensions. And as such, it is considered an extensional ontological commitment. An intensional ontological commitment, however, commits to intensional entities.

Definition; *Ontological Commitment*: According to the intensional account of ontological commitment, an intensional ontological commitment of a first order intensional logical language L_w with vocabularies V_w is an intensional structure $\mathcal{OC}(\mathcal{C}, I)$ in which I is an intentional interpretation function and $\mathcal{C}$ is a conceptualization. The intensional interpretation function I maps each vocabulary symbol of V_w to an element of D. Where $M = (D, I)$ is the standard model (intended intensional interpretation) of L_w according to the ontological commitment $\mathcal{OC}$.

In order to avoid the confusion about what an ontology is, it should be made clear that the ontology has several accounts as well (Smith 2008). Unlike the case with ontological commitment, neither of the different accounts for ontology is inadequate. However, each account is appropriate in the appropriate context. In philosophy, the term Ontology refers to a systematic account of existence. The connection between this definition and the usage of Ontology in artificial intelligence (AI) is that, for AI systems, what “exists” is what can represented (Gruber 1992). In science, ontology of a certain domain includes the terms used in this domain, and the relation between them. This ontology is developed in such a way as to be analogous to scientific theories. Such ontologies are developed and validated by domain experts to be common resources. Also, these ontologies are recognized as being always subject to further development, and are independent of format and implementation. In engineering, Ontology is a specification of a conceptualization. And thus ontologies are considered to be engineering artifacts (Gruber 1992). In this work, the ontology we are interested in is the ontology from the engineering perspective. This ontology is developed and maintained by engineers and computer scientists. This ontology will be formally defined as follows:

Definition; *Ontology*: Let $\mathcal{C}$ be a conceptualization, and L_w an intensional logical language with vocabulary V_w and ontological commitment $\mathcal{OC}$. An ontology O, for $\mathcal{C}$ with vocabulary V_w and intensional ontological commitment $\mathcal{OC}$, is a logical theory consisting of a set of formulas of L_w, designed so that the set of its models approximates as well as possible the standard model (intended intensional interpretation) of L_w according to $\mathcal{OC}$.

Note that, this definition is different from the definition that is based on the "possible world" (Guarino, Oberle, and Staab 2009). The definition that is based on the approach the "possible world" approach assumes several intended models for each language and ontological commitment. Each model is concerning one possible world. In this work, the definition of ontology follows the intensional model for conceptualization in sections 3.6.2 and 3.6.4. And as such, for a language and ontological commitment, there is only one intended interpretation. This intended interpretation (standard model) is what the ontology is required to approximate. It is also worth mentioning that, in the possible world approach, the interpretation is based on a set of possible world that may not even exist.

Chapter 4

4 Intensional Modeling of Data Integration Systems

This chapter proposes an intensional-based model for ontology-driven data integration in open environment. As described in Chapter 3, Intensional modeling is found to be more natural choice for modeling in open environment. This is due to the dynamic and loosely-coupled nature of open environment. In open environment, agents need to enter or leave the system without affecting the overall functionality. It has also been illustrated in Chapter 3 that the belief of an agent and the knowledge of an information system are intensional in nature. Formal intensional semantics for queries and query answering are then presented. The semantics presented in this chapter are based on the intensional epistemic logic (IEL) (Jiang 1993).

4.1 Introduction

One of the main issues in open environment is the heterogeneity. The issue of heterogeneity has been investigated for a long time in the information system community. In (Hull and King 1987), the importance of the semantic database modeling is discussed. Recent research has pointed out that, even though database schema contains semantics, database schemas are mainly concerned about data and data structures. Also, the semantics in the database schema are un-maintainable since they are implicit. There are also lots of hidden and implicit rules inside a database schema. And this is why it can work well with applications but not for querying the data. On the other hand, the main focus of ontologies is the meanings. An ontology does not focus on the structure of some data; rather, it describes the conceptualizations and subject matters. Ontologies also provide explicit semantics that are maintainable. These are some of the main reasons why ontologies have gained acceptance as sources of semantics.

As discussed in Chapter 3, the dynamic nature of the open environment is another challenge for modeling in open environment. There must be no constraints on the set of data sources or the number of data sources. The system must account for data sources entering or leaving the system at any point of time. This dynamic nature needs to be accounted for when modeling in open environment. It has been shown in Chapter 3 that the extensional and extensional reduction models are not adequately capable of modeling in open environment. The intensional model also overcomes the limitations of the extensional and the extensional reduction approaches in open environment.

Another challenge for modeling a Data Integration System (DIS) in open environment is loosely-coupled nature of open environment. In open environment, as there is no control over the set of data sources, there is no control over the data residing at the data sources. Even though ontology is a powerful tool for bridging the heterogeneity gab, it only deals with the heterogeneity in the semantic structure of the data. However, the heterogeneity, in the data residing at various data sources, needs to be addressed as a separate issue. This issue needs to be accounted for when modeling a DIS in open environment. And this is one of the reasons why extensional and extensional reduction approaches are ill suited for modeling in open environment. This is mainly because the belief of an agent and the knowledge of an information system are intentional matters as illustrated in Chapter 3.

In this chapter some of the challenges for data integration in open environments are discussed. These issues include the distributed nature, the dynamic nature, the heterogeneity, and the loosely-coupled nature of open environments. An intensional model for ontology-driven data integration system (ODIS) is presented. The proposed model accounts for the dynamic, heterogeneous and loosely-coupled nature of open environment. Both the mediated data integration system and P2P data integration system are discussed. It is then shown that both the mediated and the P2P architectures are special cases of the mediated P2P data integration architecture. A model for the mediated P2P ontology-driven data integration system in open environment is then presented. The semantics of the proposed model are proposed in light of the intensional epistemic logic theory.

4.2 Different Architectures for DIS

There are two major architectures for virtual data integration systems. The first is the P2P architecture Figure 5, and the second is the mediator-based architecture (Abiteboul et al. 2011) shown in Figure 2. In the P2P architecture, there is no centralized control. As such, each peer (data source) works as a DIS on its own and has to integrate with the other peers itself without the need for moderation. On the other hand, the mediator-based architecture has a mediator in the form of a global ontology or a global schema. The mediator in that case works as a coordinator and integrate the data from several data sources. While the P2P framework allows for the flexibility of querying against any peer, the mediator-based approach does not require every single information system to be a DIS.

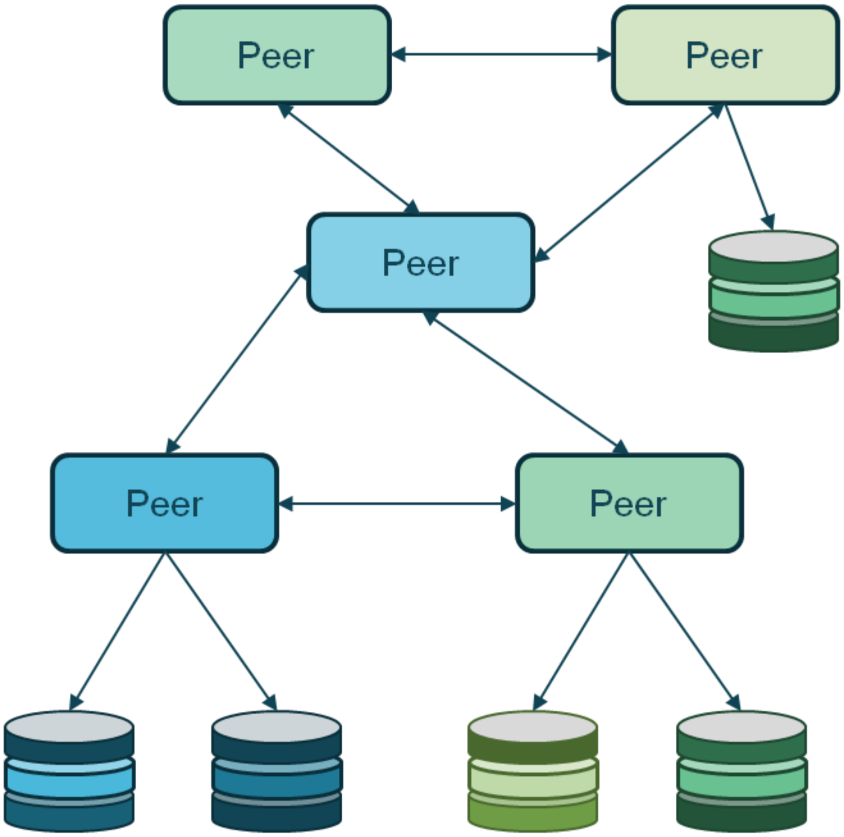

Figure 17: Mediated P2P Architecture

In open environment, it is non-realistic to expect each data source to work as a separate data integration system and integrate with the other peers without mediation. On the other hand, it is also non-realistic to assume a centralized control and force the queries to be executed against one, single global, ontology that may have to change every time a data source is added, in order to account for the new data source. The mediated P2P data integration architecture, shown in Figure 17 is found to be a compromise between the mediated architecture and the P2P architecture. In this architecture, a set of mediated networks are used and each mediated network interact with other mediated networks through a P2P interaction. This relaxes the condition that each data source needs to work as a separate DIS on its own. At the same time, the mediated P2P architecture does not require the whole system theory and global ontology to change with the addition of a new data source. As such, there can be more than one global ontology. These ontologies can talk to each other on P2P basis. Each of these ontologies can also talk to their local ontologies on mediator-based basis. The result is a hybrid model that supports both P2P and mediator-based data integration. This architecture was first proposed in (Lumineau, Doucet, and Gançarski 2006) and (Halevy et al. 2003). Using intensional epistemic logic, we will be able to formalize the mediated P2P data integration system and provide proper semantics for query answering.

There have been several attempts that try to address the problem of semantic data integration in open environment. In (Y. D. Wang 2009) an architecture is proposed which is based on web-based and multi-agent technologies. The author used ontology in order to deal with the issue of heterogeneity and focused their work on finding mapping between various ontologies in the system. A problem the authors called, finding the semantic transformation. The authors introduced the definition of ontological view, and semantic transformation techniques are used to map these ontological views to one another. The authors defined semantic transformation to be “finding a function or mapping that assigns the elements of the target ontological view vocabulary to the elements of the source ontological view vocabulary”. The author in (Y. D. Wang 2009) adopted the model described in (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009) for their modeling of the data integration systems. The author then defined the formal definitions to the following terms: Transformable, Partially Transformable, and Untransformable, to be used as a base for the ontology transformation. The author then provides a formal definition for Ontology transformation (Mapping). It is worth mentioning that the focus of the work proposed in (Y. D. Wang 2009) is the semantic transformation rather than data integration. A critical review on the adequacy of framework presented in (Y. D. Wang 2009) to open environment is detailed in Section 2.3.

In (Xue 2010), the author also presented a framework that addresses the ontology-driven semantic integration in open environment. The main focus of the work was around the semantic mapping aspect. With regards to the ontology modeling and representation issue, the author also adopted the extensional reduction model in (Guarino and Giaretta 1995), (Guarino 1998), and (Guarino, Oberle, and Staab 2009). That model is based on the "possible world" approach. It is shown in Chapter 3 that the extensional reduction model is inadequate to model information systems in open environment. The author also adopted the definition of ontological view and ontological commitment to a view from (Y. D. Wang 2009). It is worth mentioning that the author used the schema as a source of semantic. This was done by extracting data driven ontologies (DDO) from the data. This method can help in case of the sources do not have explicit ontologies. However, counting on DDO only in open environment means the schemas are the only source of semantics. And, if the schema is used as the source of semantics, this can result in inaccurate semantics. Moreover, the author used an elementary technique for ontology matching. Elementary ontology matching techniques are proven to yield less accurate results compare to structural algorithms. The author also modeled the data integrated system using a global schema that is the result of integrating several source schemas. This is what is called mediated-based data integration. The mediated-based data integration, as shown earlier, is centralized. And as such, it does not address the distributed nature of open environment.

Another framework for data integration in open environment was proposed in (Ali and Ghenniwa 2014). The model proposed in (Ali and Ghenniwa 2014) takes advantage of the intensional model for ontology and conceptualization presented in (Ali and Ghenniwa 2012). The authors created a conceptualization for the data integration system that contains four main concepts. These concepts are namely; *global ontology*, *local ontology*, *interface*, and *mapping*. The authors defined the four concepts as follows:

- Global Ontology: Is the mediator ontology which provides a unified view through which several data sources can be queried.

- Local Ontology: Is an ontology that is designed for a particular information system. If data sources do not have ontologies, there are techniques to driven ontologies from the data. An ontology driven from the data is called data-driven ontologies (DDO).
- Interface: Is defined in terms of a set of mappings between a global and a local ontology.
- Mapping: Defines a concept in one ontology in terms of concepts of another ontology. The ontological views, as defined below, describe relations between concepts of the two ontologies. In other words, a concept of one ontology is defined as an ontological view over another ontology.

The authors also expressed the data integration system as a tuple:

$$DIS = (D_{DIS}, K_{DIS}) \tag{23}$$

In which D_{DIS} is the domain of the data integration system which consists of the three sets of concepts, relations, and properties on the data integration system. On the other hand, K_{DIS} is a set of extensionalization functions as described in (Bealer 1998a).

The authors in (Ali and Ghenniwa 2014) addressed the intensional nature of open environment. However, as is the case with (Xue 2010), the proposed model is a mediated-based model which assumes centralized control in the data integration network.

As shown above, several issues for modeling data integration systems in open environment have not been adequately addressed. In this chapter, we will attempt to address these issues. We will start by shedding some light on the epistemic logic and intensional epistemic logic (Jiang 1993). And then, the intensional epistemic logic will be used to model the ontology-based data integration system in open environment. As mentioned earlier, while the P2P framework allows the flexibility of querying against any pair, the mediator-based approach does not require every single information system to be a DIS on its own. As such the proposed model will use a compromise between both the mediated and the P2P architectures.

4.3 Research Assumptions

It is natural that a proposed solution addresses a specific problem. This requires some assumptions to limit the scope of problem and, hence, reduce the complexity of the proposed solution. Below is a list of the assumption for the proposed framework:

1- All the information systems are interested in the same domain: At the representation level, the information system can use different model to represent their knowledge. However, at the conceptual level, the information systems should be built for the same world of interest.

2- For every information system, an ontology exists or can be generated: This is important because the system attempts to integrate various information system using explicit semantics. Some information system may not have a built in ontology. In order for the integration to be achievable, a data driven ontology needs to be generated.

3- The information systems have heterogeneous ontologies: This is natural since the information systems in open environment are expected to be designed by different parties with different objectives in mind.

4- The mappings between ontologies exist: The proposed model is built with the assumption that elements of various ontologies map to one another. Since all the information systems are built for the same domain, it is reasonable to assume that mappings between the elements of their ontologies exist.

5- The matching between elements of various ontologies is achievable: The proposed model utilizes the mappings between the elements of various ontologies to transform queries over the ontology of one information system to a query over the ontology of another information system. This enables the system to use one information system to answer queries posed to another information system. It is assumed that mappings between the elements of various ontologies exist. As such, it is reasonable to assume that, it is possible to find these mappings through an ontology matching algorithm.

4.4 Propositional Epistemic Logic

Epistemic logic is the logic of knowledge and belief. Even though, epistemic logic and doxastic logic formalize the knowledge and belief, respectively, the term epistemic logic is also commonly used to refer to the both the logic of knowledge and the logic of belief. The main focus of epistemic logic is the propositional knowledge. That said, an agent bears the propositional attitude "knowing" or "believing" towards a proposition. As such, when we say: "Joe knows that Tom loves Merry" we are asserting that Joe is an agent who bears the propositional attitude "knows" towards the proposition expressed by "Tom loves Merry".

The syntax of the propositional epistemic logic is simply the result of augmenting the language of propositional logic with the unary knowledge or belief operators K_a or B_a. where a is an agent, and the operators K and B are the epistemic operators for knowledge and belief respectively. In that sense, if P is an arbitrary proposition, following is how these operators are read:

K_aP reads "Agent a knows that P"

And similarly for the belief operator in the case of doxastic logic:

$B_a P$ reads "Agent a believes that P"

4.5 Intensional Epistemic Logic

As discussed earlier, the knowledge and belief are intensional contexts. Intensional epistemic logic offers a way to properly handle relative intensions in nested believes. The most distinguished feature of the intensional epistemic logic is the use of intensional index on the terms. The basic idea is that, given a formula like $B_a\, p(b)$, b does not have to be rigid. That means, b does not have to have the same meaning everywhere in the formula or same denotation in all possible worlds. And so, we need some mean to distinguish the case when b is evaluated inside the intensional scope of agent a, and the case when b is evaluated outside the intensional scope of agent a. To achieve this, a superscripted index is attached to each term to denote the number of the believe operator that contains the intended meaning of the term. If a term is not attached with an intensional index, then the intended meaning of the term is rigid. The following example uses the superscripted terms to capture intensionality:

Example: consider the formula $B_a(Q \wedge B_bQ)$; if Q's intended meaning is in the scope of B_a, the formula can be represented in IEL as $B_a(Q^1 \wedge B_bQ^1)$. On the other hand, if the second Q in the original formula is intended to be local to B_b, then the formula should be represented in IEL as: $B_a(Q^1 \wedge B_bQ^2)$.

As such, the language for intensional epistemic logic is a first order logic language with equality, augmented with the believe operator *B* for each agent, with superscripted terms.

4.6 Intensional Model for DIS in Open Environment

In this section logical framework for ontology-driven mediated-P2P data integration in open environment is proposed. The mediated P2P architecture adopted here was first proposed in (Lumineau, Doucet, and Gançarski 2006) and (Halevy et al. 2003). The intensional epistemic logic is also utilized in order to present formulations and semantics for the application of mediated P2P data integration in open environment.

In the proposed formulation there is no need to assume that there is only one domain for all the data sources or mediators in the network. Instead, several domains can be considered. This makes sense when the system is dynamic and loosely-coupled. And, with the use of intensional epistemic logic, proper semantics and explanation for this are provided. As such, using intensional epistemic logic, a query or a term does not have to have the same interpretation or denotation in all possible worlds. Attaching a superscripted index to the term or the query will indicate the number of the belief operator that will include the intended meaning or the intended interpretation of the term or the query. Another main feature of the proposed model is that, the answer to a query does not have to depend on the satisfaction of the query in a universal model of the whole P2P system. Instead, every mediator network will be treated as a separate entity and the answer to the query would be the union of all the answers coming separately from each mediated network, independent of what other mediators believe.

In this formulation, a mediated P2P DIS will be modeled as a two level logic system. Each level will be formulated as a set of intensional epistemic logic theories. The first level is the P2P level which will model the interaction between various mediators for the purpose of answering a user query over one of the mediators' ontologies. The second level will be a mediated level that will model the interaction inside the local network of the mediator. The main reason why the model is divided into two levels is to distinguish between the theory of one peer, a mediator, and the theory of the P2P system. This will abstract out the structure of one mediated network and the interaction that will happen within the mediator's network. More importantly, as has been discussed earlier, the open environment is dynamic in nature. And as such, when we separate the model into a P2P level and a set of mediated levels for each peer, the addition of a data source and the withdrawal of a data source are abstracted out. As such, this will not affect the logic theory or the interaction at the P2P level or other mediators' networks. This will create a modular and scalable system that is dynamic and can serve the objectives of an open environment adequately.

Since the query can be asked to any peer, we model the mapping in the P2P network as Global-Local-As-View GLAV mapping (Friedman, Levy, and Millstein 1999). On the other hand, within the mediated network of each peer, the query will always come from the peer, or the mediator of the peer in other words, to a data source in the peer's local network. As such, the mapping will be modeled as Global-As-View GAV mapping (Lenzerini 2002). This will facilitate the query answering in both the P2P network and the local peer's mediated network.

In one view for modeling the data integration system, the whole network is formulated as a single FOL theory. In doing so, the whole network is thought off as a single integrated entity. In practice, in a distributed system in open environment, a peer does not interact with other peers that are not directly connected to it. As such, the peer cannot distinguish the status of other peers other than its immediate neighbors. In order to account for this characteristic of open environment, a set of distinguished theories are going to be considered. Each theory will be an intensional epistemic logic theory that is only concerned about one peer and its immediate P2P or mediated networks. That said, in a mediated P2P network with *N* peers, there will not be a single theory that represents the entire network as a whole. Instead, reasoning will take place in stages and each stage will be represented by a separate IEL theory.

Definition: Mediated P2P data integration system; an ontology based Mediated P2P data integration system of *N* mediated peers in open environment is defined as:

$$MP2P = \{MP_i | 1 \leq i \leq N\} \qquad \textbf{(}24\textbf{)}$$

where MP_i is a mediated peer network defined as a tuple:

$$MP_i = (OP_i, OG_i, S_i, R_i, G_i, L_i) \qquad \textbf{(}25\textbf{)}$$

where:

OP_i: is the private ontology that is local to the mediated peer MP_i and is not accessible to other mediated peers MP_j in the P2P level of the mediated P2P network.

OG_i: is a global ontology for the mediated network MP_i that is shared with the immediate P2P neighbors of MP_i. Note that there is no mapping between the two ontologies. Any query over OG_i is a query over OP_i. The purpose of OG_i here is to represent the autonomy of each peer in a way that it decides what to share with other peers and what to hide. As such, all concepts and relations of OG_i are also elements of OP_i and not the opposite. The following relationship holds between the private ontology and the global ontology:

$$OG_i \subseteq_O OP_i \qquad \textbf{(}26\textbf{)}$$

Where:

The operator $\subseteq_O$ is understood as; any query that can be answered by ontology OG_i can also be answered by OP_i.

S_i: is a set of data sources for the mediated peer MP_i.

R_i: is a set of accessibility relations between the mediated peer MP_i and other mediated peers in the P2P network.

G_i: is a set of P2P interfaces G_{ij}, each of which consists of a set of mappings between the elements of the private ontology OP_i of the mediator peer MP_i and elements of the global ontology OG_j of its neighboring mediator peer MP_j. Each mediated network will have global mappings between the concepts of its own private ontology and the global ontology of other P2P neighboring mediated networks. The ontological views, as defined below, describe relations between elements of the two ontologies. In other words, a concept of one ontology is defined as an ontological view over another ontology.

Definition; *Ontological View*: an ontological view over an ontology is a stored query over that ontology.

In the general form, the mapping takes the following form:

$$q_i(x) \rightsquigarrow q_j(x) \qquad \textbf{(}27\textbf{)}$$

The mapping above maps an ontological view over the local ontology OP_i to another ontological view over the global ontology OG_j.

Note that, the head of the arrow of equation (27) is an ontological view over the global ontology of mediated peer MP_j. This is because mediator peer *MPj* may not share its private ontology with other peers in the network. Rather, what is shared is only the global ontology which is related to its private ontology with the relationship in equation (**26**). This is to preserve the autonomy of each mediated peer by allowing each peer to decide what to share with other mediator peers in the P2P network.

L_i: is a set of sets of local mappings L_{ik}. Each L_{ik} is a set of local mappings between the concepts of the private ontology OP_i of the mediated peer MP_i and the local ontologies of the data source $S_{ik} \in S_i$ where S_i is the set of local data sources for the mediated peer MP_i. If data sources do not have ontologies, there are techniques to driven ontologies from the data. An ontology driven from the data is called data-driven ontologies (DDO).

In traditional data integration systems, the whole data integration network is formulized by a single theory that represents all the data sources involved in the network. While this approach can be useful in certain situation, it is not the proper way to formulize a dynamic distributed system in open environment. Intuitively speaking, when we deal with a distributed system, if a query is posed to the private ontology OP_i of a mediated peer MP_i, the answers to the intensionally equivalent query that is executed against another peers will be considered as part of the global answer to the original user query. However, these answers are based on the relative believes of each mediated peer about the knowledge of its own neighboring mediated peers. As such, mediated peer MP_i can only make claims about what it beliefs the knowledge of its own neighboring peers is. Those neighboring mediated peers of MP_i can make claims about their own believes regarding the knowledge of their own neighboring mediated peers and so on. As such the global answer will be expressed in terms of the nested believes and will be calculated in stages until the last mediated peer is reached. This shows that the whole network in the intensional epistemic logic setting is not formulized as a single theory. Rather every mediated peer and its immediate neighboring network are represented by a separate theory. At the same time, the mediated network of each mediator peer has its own intensional epistemic logic theory as well.

Definition: The formalization of a mediated P2P data integration system: The ontology based mediated P2P data integration system in open environment is formalized as a set T_{GP} of N distinguished global IEL theories, one for each mediated network MP_i, and a set T_{LP} of N distinguished local IEL theories, one for each mediated network MP_i. This can be expressed as:

$$T_{MP2P} =< T_{GP}, T_{LP} > \qquad (28)$$

With:

$$T_{GP} = \{T_{GPi} | 1 \leq i \leq N\} \qquad (29)$$

And

$$T_{LP} = \{T_{LPi} | 1 \leq i \leq N\} \quad \textbf{(}30\textbf{)}$$

The 2N logical theories defining the mediated P2P network are defined as follows:

Each global, P2P, intentional epistemic logic theory T_{GPi} is defined by:

1- The set *AGTS* of agents for each intensional logic theory T_{GPi} for the mediated network MP_i is the union of the $\{P_i\}$ and the set of its immediate neighboring mediated peers $R_i(MP_i)$. We abuse the notation by using $R_i(MP_i)$ to mean the set of all immediate P2P neighbors to the mediated network MP_i.

$$AGTS = \{P_i\} \cup \{P_j | MP_j \in R_i(MP_i)\} \quad \textbf{(}31\textbf{)}$$

Where R_i is the set of accessibility relations for mediated peer MP_i.

2- The alphabet A_{TGPi} for the intensional epistemic logic theory T_{GPi} is the disjoint union of the alphabets of the private ontology *OPi* of the mediated peer *MPi* and the alphabets of the global ontologies OG_j of its immediate P2P neighboring mediated peers.

$$\mathcal{A}_{TGPi} = \mathcal{A}_{OPi} \sqcup \{\mathcal{A}_{OGj} | MP_j \in R_i(MP_i)\} \quad \textbf{(}32\textbf{)}$$

3- All the formulas of the private ontology OP_i of the mediated network MP_i and the global ontologies OG_j of its immediate P2P neighbors are going to be axioms in the theory T_{GPi}.

4- For every global mapping assertion in the set G_{ij} of the form:

$$q_1(x) \rightsquigarrow q_2(x) \quad \textbf{(}33\textbf{)}$$

there is an axiom in T_{GPi} in the form:

$$\forall x (B_{Pi} B_{Pj} q_1(x)^2 \leftarrow B_{Pj} q_2(x)^1) \quad \textbf{(}34\textbf{)}$$

Which is interpreted as; if mediated peer MP_j believes something about the query $q_2(x)$, then the neighboring P2P mediated peer MP_i believes that peer MP_j believes the same thing about the query $q_1(x)$ evaluated at mediated peer MP_j. Here query $q_1(x)$ evaluated at peer MP_j is understood to be the result of applying the appropriate P2P mappings to $q_1(x)$ to yield a query $q_2(x)$ over the global ontology of mediated peer MP_j and executing the query $q_2(x)$ to get the answer in an actual interpretation at mediated peer MP_j.

On the other hand, each local, mediated, intentional epistemic logic theory T_{LPi} is defined by:

1- The set *AGTS* of agents for each intensional logic theory T_{LPi} for the mediated network *MPi* is the union of the set $\{Pi\}$ and the set *Si* of all data sources of the mediated network *MPi*.

$$AGTS = \{P_i\} \cup S_i \tag{35}$$

2- The alphabet A_{TLPi} for the intensional epistemic logic theory T_{LPi} is the disjoint union of the alphabets of the private ontology OP_i of the mediated network MP_i and the alphabets of the set S_i of its local data sources.

$$\mathcal{A}_{TLPi} = \mathcal{A}_{OPi} \sqcup \{\mathcal{A}_{Sik} | S_{ik} \in S_i\} \tag{36}$$

3- All the formulas of the private ontology OP_i of the mediated network MP_i and the ontologies of all data sources of the local mediated network MP_i are going to be axioms in the theory T_{LPi}.

4- For every local mapping assertion in the set L_{ik} of the form:

$$q_1(x) \rightsquigarrow q_2(x) \tag{37}$$

there is an axiom in T_{LPi} in the form:

$$\forall x(B_{Pi} q_1(x)^1 \leftarrow q_2(x)) \tag{38}$$

> Which is interpreted as; if there is an assignment that makes query $q_2(x)$ true in the intended interpretation of data source S_k of mediated network *MPi*, then MP_i believes the same thing about the intensionally equivalent global query $q_1(x)$. Here query $q_2(x)$ is the result of applying the appropriate local mappings L_{ik} to $q_1(x)$ to yield a query $q_2(x)$ over the data source S_k.

Note that, the mediated P2P network is formalized as a set of *2N* intensional logic theories instead of one single theory. Each global theory T_{GPi} considers the P2P mappings between the mediator peer MP_i and its immediate P2P neighbors, but does not consider any local mappings or even other P2P mappings in the network. On the other hand, every local theory T_{LPi} of mediated network MP_i considers only its local mappings L_i but does not consider any P2P mappings. The mediated P2P network in this setting can be seen as a set of collaborating data integration systems. Each data integration system is consists of a peer, the set of its neighboring peers, and the set of the local data sources of its own mediated network. As such, each, one of these collaborating, data integration systems is formulated with a two level intensional epistemic logic theory. The first level is an epistemic intensional logic theory which involves a mediator, its immediate P2P neighboring mediators and its global P2P mappings. The second level, however, is another intensional epistemic logic theory that involves one mediator, all the data sources in its own local network, and all the local mappings.

And so, the answer to a user query $q(x)$ posed to the private ontology OP_i of a mediated network MP_i is equal to the extensions of the query $q(x)$ in the data sources of mediated network MP_i union the answers to the equivalent queries evaluated at the immediate neighbors $R_i(MP_i)$. The answer to the query evaluated at the immediate neighbors of the peer, to which the query is posed, is also going to be calculated in terms of its local extensions, union the answers to the query evaluated at its own immediate neighbors. And so on until the last peer is reached. In other words; the belief of mediated peer MP_i about a query $q(x)$ consists of its own local beliefs union what MP_i believes its neighbors believe about the intensionally equivalent queries. And, this is one of the advantages of IEL is it offered a language that can clearly describe these settings.

It is also important to note that, the term “Agent” is commonly used in the collaborative intelligence context to refer to an entity that is autonomously contributing to a problem solving network. For the sake of this work, the term agent is used to refer to an information system participating in a data integration system network. The problem at hand is to answer queries. Each information system contributes the solution of the problem by providing an answer to the intensionally equivalent query. The autonomy of each information system reflects in their choice to associate or dissociate with the environment. Also, the fact that each information system decides on what to share/hide with the rest of the network is another factor that reflects the autonomy of information systems.

4.7 Interface and Query Semantics

The interface between the global ontology and the data sources is calculated in terms of mapping. Depending on whether a local-centric or a global-centric model is used, the interface maps concepts of one ontology to queries over another ontology. Both the local-centric and the global-centric models can be considered as special case of the GLAV model (Friedman, Levy, and Millstein 1999). In the GLAV model, queries of one ontology are mapped to equivalent queries over another ontologies. This mapping requires the two queries to be equivalent. There are two different ways the equivalence between two queries can be viewed:

Extensional equivalence: Two queries are considered to be extensionally equivalent if the data sets returned from executing the two queries are identical.

Intensional equivalence: Two queries are considered to be intensionally equivalent if they are expressed in terms of equivalent intensional entities.

The extensional equivalence is inappropriate to describe equivalence in open environment. This is because, in this type of environment, several agents can have diverse knowledge or beliefs about the same concept. And in turns, they may have different beliefs about equivalent queries as well. This also makes extensional equivalence very likely to be unattainable. It is also worth mentioning that queries themselves can be viewed as intensional matters as they are expressed in terms of intensional entities. Also, as illustrated in (Bealer 1979), the reductionist approaches, i.e. possible-world approaches, are based on a conception that is ideally suited for treating logical modularity. However, this conception has proven to be of little value for describing intensional matters such as belief, desire, perception, decision, assertion, etc. For these reasons, the intensional equivalence is chosen to describe the equivalence between different queries. This is necessary in order to map queries over one ontology to equivalent queries over another ontology. The intensional equivalence however does not imply extensional equivalence. The following example from (Bealer 1979) explains this issue. Consider the following invalid argument involving the intensional predicate ‘Wonders’:

x wonders if there is a trilateral that is not a triangle.

Necessarily, all and only trilaterals are triangles.

∴ ***x wonders if there is a triangle that is not a trian1gle.***

This argument is intuitively invalid. In (Bealer 1979) it is argued that, even though necessary equivalence is necessary for identity, it is not sufficient. The intensional entities need to have a unique and non-circular definition.

Definition; *Intensional Equivalence of Queries*: Two ontological views $q_1(x)$ and $q_2(x)$ are intensionally equivalent ($q_1(x) \equiv_{in} q_2(x)$) if they are expressed in terms of intensionally equivalent concepts.

When dealing with data integration in open environment, it is important to note that, the answer to a query does not needs to be true in all models of the network before it is delivered from one peer to another. It is sufficient for the answer to be true in the intended model of the network of the peer that is answering the query. This is because we do not assume a single theory for the entire network. Rather, the proposed model assumes a set of collaborating networks. Assuming a single theory for the entire network is not realistic given the nature of the distributed systems in which there is no centralized control. As such, there will be no unified view of all the mediated networks in the P2P network. Instead, each mediator peer will be visible to the mediator peers that have immediate access to it only. And so, each mediator peer will only receive queries from and deliver answers to those peers that have direct access to it. In order to deal with this issue, the answer to a query posed to a peer is expressed in terms of the local belief of this peer and the nested believes of the peers that are accessible from this peer using the accessibility function *R* defined above. In doing so, a mediator peer MP_i does not claim any knowledge or belief about the belief of another mediator peer MP_k unless mediator peer MP_k is directly accessible to mediator peer MP_i. As such, if there is still a connection between two peers, that is not immediate, this will be described in terms of nested belief. For example, if mediator peer MP_i is connected to mediator peer MP_k through mediator peer MP_j, mediator peer MP_i does not make any claim about the beliefs of the mediator peer MP_k. Rather, mediator peer *MPi* can claim its belief about the beliefs of mediator peer MP_j. MP_j can then make claims about the beliefs of mediator peer MP_k since MP_k is directly accessible from mediated peer MP_j. As such, mediator peer MP_i can claim that it believes that mediated peer MP_j believes that mediated peer MP_k believes something, and so on.

This will be described in intensional epistemic logic in which the intensional index will be used to indicate the belief operator, and in turns the domain, in which the query will be evaluated. There will be a number *N* of separate domain, one for each mediated network, and the intensional index of the IEL will be employed to determine the domain in which the query is being evaluated. Also, when describing the semantics of the mediated P2P network, we assume that all the data sources of each mediated network are federated into one integrated data source. In doing so, describing the semantics in the P2P level will be made easy to understand.

Definition: *The intensional semantics of a mediated P2P data integration system*: The semantics of the ontology based mediated P2P network in open environment can be described as follows:

We consider a model $\mathcal{M}$ for the intensional epistemic logic ontology driven mediated P2P data integration network of N mediated networks as a structure $\mathcal{M} =\langle W, \pi, D, \mathcal{K}\rangle$, where:

W: is the set of the different states or interpretations for the mediated P2P network. Here we limit the set of possible interpretations to the actual interpretations, intended interpretation, at each mediated network.

π: is a set of reflexive relations on the form (w_{ik}, w_{ik}) where w_{ik} is a possible states for the mediator peer MP_i and $(w_{ik} \in W)$. As such, it is enough for the query to be satisfied in the actual world in order for the extensionalization of the query to be an answer. This type of relation also indicates that each mediator peer cannot distinguish the cases where the states of other peers changed if its own state does not change.

$D = \{D_1, D_2, \dots D_N\}$ is the disjoint union of the domains of all the mediator in the network.

$\mathcal{K}$: is a set of extensionalization functions for the mediators. It follows that, for a query $q(x)$ posed to a mediator peer MP_i, the local answer to the query is $\mathscr{k}_i(q_i(x)) \in D_i$. The global answer includes all the answers for the equivalent queries $\mathscr{k}_j(G_{ij}(qi(x))) \in D_j$ for each mediated network MP_j accessible to mediated network MP_i and so on.

A query $q(x)$ is satisfied in a state w_{ik} of a mediator peer MP_i by the tuple of constants c $\mathcal{M},w_{ik} \vDash_c q(x)$ if $\mathscr{k}_j(q(x)) = c \in D_i$ and $q(c)$ is true in interpretation w_{ik} of peer MP_i. Where $\mathscr{k}_j(q(x))$ is the extensionalization of query $q(x)$ in the world w_{ik} of a mediator peer MP_i.

An atom of the form $B_{Pi}(q(x)^1)$ is satisfied in the world w_{ik} of mediator peer MP_i by the tuple of constants c, $\mathcal{M}, w_{ik} \vDash_c B_{Pi}\ q(x)^1$, if $q(c)$ is true in state w_{ik} of mediator peer MP_i and $\mathscr{k}_i(q(x)) = c \in D_i$. Note that, this is equivalent to saying that $q(c)$ is true in all world w_j where $(w_i, w_j) \in \pi$, however, it will yield the same result since π is only a reflexive relation which means that the set of possible worlds for peer MP_i is a set of only one member which is the actual world w_{ik} for mediator peer MP_i

An atom of the form $B_{Pi}\, B_{Pj}(q(x)^2)$ is satisfied in the mediator peer MP_i by the tuple of constants c, $\mathcal{M}, P_i \vDash_c B_{Pi}\, B_{Pj}\, q(x)^2$ if mediator peer MP_j is accessible from mediator peer MP_i and $G_{ij}(q(c))$ is true in a world w_{jl} of mediator peer MP_j and the extensionalization of query $G_{ij}(q(x))$ in the world w_{jl} of mediator peer MP_j is $ƙ_j(G_{ij}(q(x)))$ $= c \in D_j$. where $G_{ij}(q(x))$ is the result of applying the global mapping G_{ij} to the query $q(x)$.

An atom of the form $B_{Pi}\, B_{Pj}\, \ldots\, B_{Pm}(q(x))$ with n nested modal belief operators is satisfied in the actual world of peer *MPi* by the tuple of constants c if $B_{Pj}\, \ldots\, B_{Pm}(G_{ij}(q(x)^{DEC}))$ is satisfied in a possible world of mediator peer MP_j by the tuple of constants $c \in D_j$. Here $q(x)^{DEC}$ is the result of decreasing all the intensional indexes in the formula $q(x)$ by 1.

As mentioned earlier, the answer to a query posed to a mediated network MP_i in a mediated P2P network in open environment is the union of the local answer in the mediated network MP_i and the answer to the equivalent query at the mediated networks directly accessible from the peer mediated network MP_i. As such, this can be represented as an acyclic tree with a root node representing mediated peer MP_i in the first level. The second level will have nodes representing the peers directly accessible from peer network MP_i and so on. The construction of the tree will continue until a branch of the tree hits a node that does not have access to any other nodes or a node that is already on the branch. In doing so, the tree is guaranteed to be acyclic and to contain all the paths from the root to all accessible nodes from the peer to which the original query was posed. The following section provides an example to such tree.

4.7.1 Tree-Based Query Answering

The flexibility of a mediated P2P data integration system allows for the queries to be posed to the private ontology of any of the mediator peers in the network. However, there is no guarantee that there is a path, and in turns a possible answer, from the peer to which the query is posed and all other mediator peers in the network. Moreover, because of the autonomous nature of the mediator peers in open environment, even if such path exists, there is no guarantee that the answer, if exists, will be shared with the peer to which the query is posed. As such, it is important to compute all the possible answers in the mediated P2P network. In order to retrieve all the possible answers in a mediated P2P network, a tree-based technique is proposed here to address this issue. The mediated P2P network, here, will be reduced by abstracting away the local network of each peer. The network is then represented by a graph with a set of nodes, corresponding to the set of mediator peers, and a set of arcs, representing the accessibility relations between the peers. Because there is no guarantee that the graph is acyclic, the proposed technique will have to account for a cyclic network and returns a tree that is based on an acyclic graph. The proposed technique attempts to find all possible paths from the mediator peer, to which the query is posed, to all the accessible nodes. This approach is particularly useful when satisfying the query is the only metric to consider (Yang and Garcia-Molina 2002). The definitions of the global answer, the set of all possible answers, and the local answers, at a certain data source, are then provided

Consider the mediated P2P network in Figure 18. For simplicity, only peers are displayed in Figure 18, while the local data sources of each mediated network are abstracted out. In order to calculate all the possible answers to a query posed to the mediator peer P_1 in the mediated P2P network shown in Figure 18, the tree in Figure 19 is constructed.

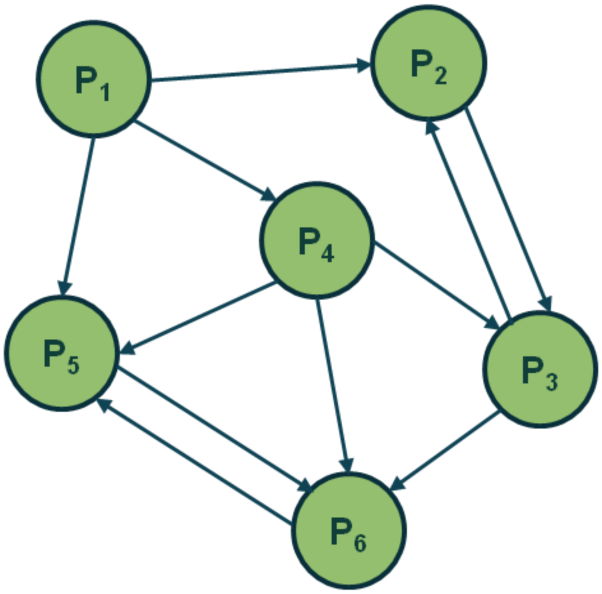

Figure 18: Example mediated P2P network

Note that, in Figure 19, the same node can appear in the tree more than once. Note also that the same node can appear in the same level more than once. This will depend on how many routes exist from the peer, represented by the root node, to the peer represented by this node. We prefixed the nodes in the levels, past the second level, in order to make it more readable. Calculating all the possible answer to a query posed to the peer P_1 in the network is equivalent to calculating the answers at all the nodes of the tree in Figure 19. This assumes that there is some mappings exist from the root, all the way, to the node at which the query is calculated.

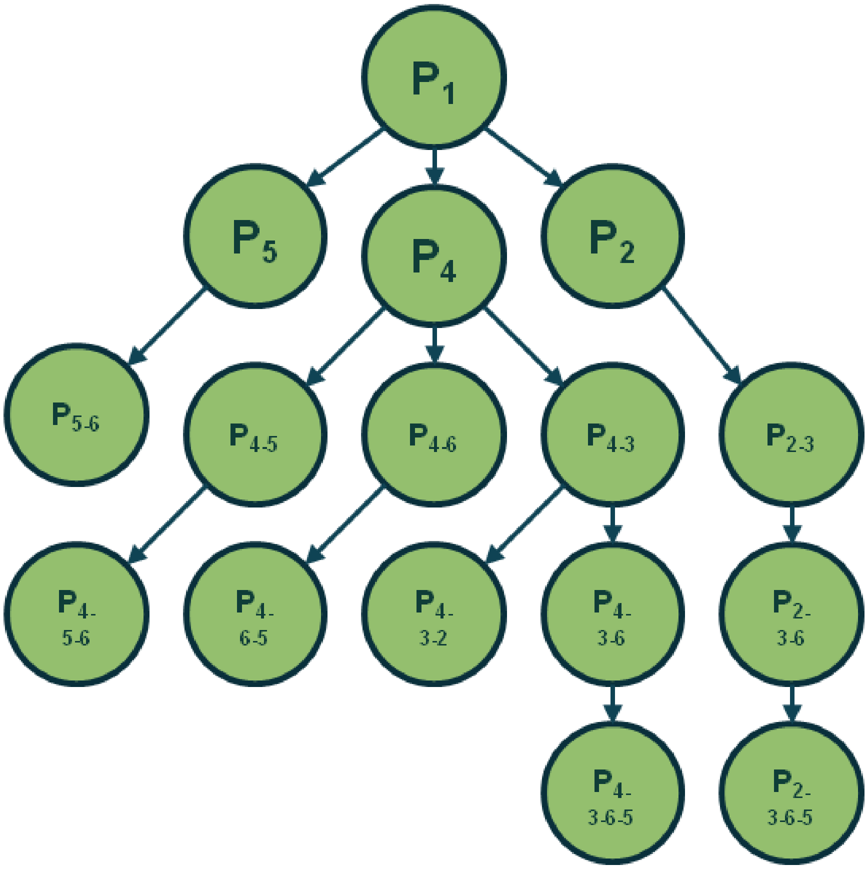

Figure 19: Acyclic Query Answering Tree for the Network in Figure 18

The algorithm for generating the acyclic tree, like the one shown in Figure 19, takes the graph representing the mediated P2P network and returns the corresponding query answering tree based on the acyclic version of the graph. The acyclic version of the graph is attained through a step in the algorithm that ignores the child if it has already appeared in the list of ancestors of the parent. The algorithm for generating the tree is shown in Figure 20.

```
Generate-Acyclic-Query-Tree-From-Cyclic-Graph
Inputs: Cyclic Graph, A Peer $P_i$, A Query Q(x)
Outputs: Acyclic-Query-Tree
Begin
    Create the root of the tree with label $P_i$
    Push the root into the Stack
    While the Stack is not empty
        Pop from Stack into x
        For each label $\ell \in R$(x) in the original graph
            If $\ell \notin$ all-ancestor-labels(x) in the tree AND
              A Mapping $G_{label(x),\ell}$(Q(x)) Exists
                Create a node y
                Q(y) = $G_{label(x),\ell}$(Q(x))
                Push y into Stack
                Add y to children(x) in the tree
            End
        End
    End
End
```

Figure 20: The Algorithm for Generating the Acyclic Query Tree

Given the tree in Figure 19, according to the proposed semantics, the global answer to a query $q(x)$ posed to a peer P_i is expressed in term of the set of all possible answer to the query. If we refer to the global answer as Ans_g and the possible answers as Ans_p, the global answer for $q(x)$ at P_i is expressed as follows:

$$Ans_g\,(q(x), MP_i) = B_{Pi}\, q(x)^1 \cup Ans_p\,(q(x), P_i, P_i) \tag{39}$$

Where:

$$B_{Pi}\, q(x)^1 = \cup_{k\in Si}\ \text{ꝁ}_{ik}(q(x)) \tag{40}$$

And

$$Ans_p\,(q(x), P_i, P_i) = \cup_{\,j\in Children(Pi)}\, Ans_p\,(q(x), P_i, P_j) \tag{41}$$

And

$$Ans_p\,(q(x), P_i, P_j) = B_{Pi}B_{Pj}\, q(x)^2 \cup Ans_p\,(G_{ij}(q(x)), P_j, P_j) \tag{42}$$

The global answer to the query is the set of all possible answers in the query tree in a nested manner. In that sense, the beliefs of a nodes about a query affects the beliefs of all its ancestors about the equivalent queries but not the other way around. As such, the beliefs that a leaf node has about the query will affect its parent and so on until the root of the tree is reached.

Given a mediated P2P query tree as the one shown in Figure 19, a data source S_{jk} in the mediated network of mediator peer MP_j, and a query $q_j(x)$ over the global ontology of mediator peer MP_j. the local answer to the query $q_j(x)$ will be referred to as Ans_l and will be expressed as follows:

$$Ans_l(q_j(x), P_j, S_{jk}) = B_{Pj}B_{Sjk}\, q_j(x)^1 = \text{ꝁ}_{jk}(q_j(x)) \tag{43}$$

Where $\text{ꝁ}_{\mathrm{jk}}(q_j(x)$ is the extensionalization of the intensionally equivalent query to $q_j(x)$, after applying the proper local mapping L_{jk} between the ontology of mediator peer MP_j and data source S_{jk}.

The following chapter will provide a case study to illustrate how to utilize the intensional logic and the intensional model for data integration systems in open environment.

Chapter 5

5 Case Study and Analysis

This chapter describes an example of applying the intensional logic and the intensional model to a real world example. The example uses the University-department world. Figure 21 shows a snapshot of the world with some professor student relations. The plan is to test seve1ral factors leading to the formal treatment of the conceptualization and ontology. We will create several ontologies for the same domain, that are not compatible with one another, and we will use them in order to describe the data integration in open environment.

5.1 Problem Specification

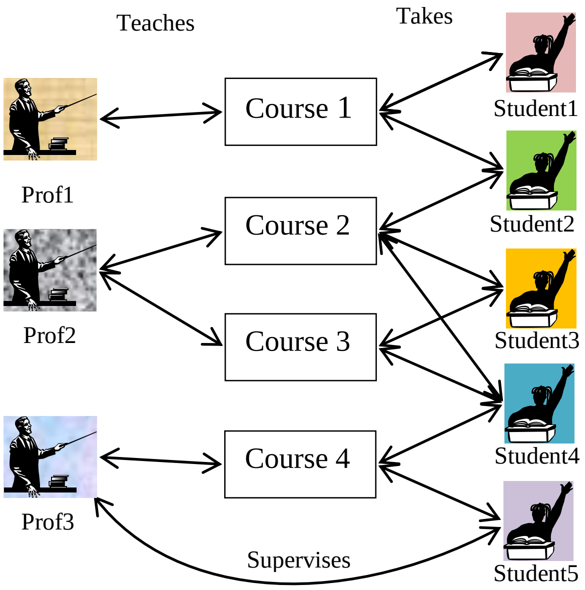

Figure 21: Example of a University-Department World

In this section we will start by considering a real world example and then, from there, discuss how the conceptualization and ontology is derived from this real world example. The example we will use is a university-department world that has some professors, some students, some courses, and the relations between them. At certain point of time, a snapshot of the world is shown in Figure 21. The snapshot of the world shown in Figure 21 shows three professors (Professor 1, Professor 2, and Professor 3), four courses, (Course 1, Course 2, Course 3, and Course 4), four undergrad students (Student 1, Student 2, Student 3, and Student 4), and a graduate student (Student 5). As can be seen in Figure 21, a professor can teach a course and supervise a graduate student. A student can take a course that is offered by the department and is being taught by a professor. A course is taught by a professor and is taken by a student. For example, Student 1 and Student 2, in Figure 21**,** both take Course 1 which is being taught by Professor 1. In order to be able to manage and reason about the information in this world, there has to be a way to represent the knowledge perceived in this world. Before representing the knowledge, a conceptualization for the world needs to be created. As mentioned earlier, there have been several proposals for the formal treatment of conceptualization which were discussed in Chapter 3. These proposals differ based on their definition of conceptualization and the class of logic their formulation of conceptualization is based on. In order to model in open environment, there needs to be a method by which we account for the dynamic nature of such environment. As has been discussed earlier, the extensional model is based in the extensional logic. As such, it describes the conceptualization in terms of declarative sentences and ordinary relations. And so, it is good for representing a snapshot of the world but does not adequately model a conceptualization in a dynamic open environment. On the other hand, the extensional reduction model overcomes some of the limitations of the extensional model. It does that capturing the changes in the relations between the world's entities, given the set of participating entities remain the same. This model is good for describing a static system with fixed set of participating entities. The intensional model however, is based on intensional logic. It does not capture the entities in the world. Rather, it captures the concepts, conceptual relations, and properties of the concepts in a given world. This enables the intensional model to allow for any changes in the world as long as the changes do not change the meanings. Below, the various formal methods for treating the conceptualization and ontology will be described in light of the example presented in Figure 21**.**

The models we are going to examine are the three models presented in Chapter 3, namely; the extensional model, the extensional reduction model, and the intension model. The following factors will be compared in the three models: the *entities*, the *concepts*, the *possible worlds*, the *relations*, the *properties of the concepts*, and the *domain*. Afterwards, the *conceptualization* and *possible ontology* will be described and compared in light of the factors listed above.

1- *The Entities*: Both the *extensional* and the *extensional reduction* models capture the entities in a given world similarly. On the other hand, the intensional model does not capture the entities as they are extensions and the intensional model does not capture extensions. Rather, the intensional model captures kinds or classes. The set of entities according to the three models is:

    a. *Extensional Model*: {*Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

    b. *Extensional Reduction Model*: {*Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

    c. *Intensional Model*: Does not capture extensions.

2- *The Concepts*: The extensional model is based on extensional logic. As such, it does not capture concepts. The extensional reduction model, however, treats the concepts as unary relations. As such, classes or kinds are captured as relations and are grouped with the relations as will be described below. On the other hand, the intensional model is the only model that captures concepts as primitives through the perception of its extensions. The set of concepts according to the three models is:

    a. *Extensional Model*: Does not capture concepts

   b. *Extensional Reduction Model*: Does not capture concepts, instead, treats classes of extensions as unary relations

   c. *Intensional Model*: $C = \{Professor, Course, Student ,GradStudent\}$

3- *The Possible Worlds*: The extensional reduction model is based on the possible world approach. As such, it captures what is called possible worlds. The possible worlds represent different arrangements of the entities in the actual world. The extensional and the intensional models are not based on the possible world approach. And as such, they do not capture the possible worlds when formally treating a conceptualization. The set of possible worlds in the three models is:

   a. *The Extensional Model*: Does not capture possible worlds

   b. *The Extensional Reduction Model*: $W = \{w_1, w_2, \ldots.\}$

   c. *The Intensional Model*: Does not capture possible worlds

4- *The Relations*: The relations in a given world are treated very differently in the three models. In the extensional model, the relations are ordinary extensional relations that exist in a snapshot of the world. As such, to the extensional model, the relations are extensional relations between entities that exist in a snapshot of the world. The extensional reduction model, however, treats the relations based on the possible-world to which the relations belong. As such, the relations in a snapshot of the world are very similar to the extensional relations. The significant difference between the relations in the extensional model and the extensional reduction model is that the extensional reduction model captures the extensional relations in all possible worlds. Whereas, the extensional model captures the relations in one world, that is the actual world. It is also worth mentioning that the extensional reduction model treats the classes of objects as unary relations and group them with the other relations in the world. The intensional model, however, captures the intensional relations in the world rather than the ordinary relations. The intensional model also distinguishes between the classes of objects and the relations. As shown earlier, the classes of objects are captured as concepts in the intensional model. Following is the set of relations for the three models derived from Figure 21:

a. *The Extensional Model*: *R*={ *Teaches(Professor1,Course1), Teaches*(*Professor2,Course2*), *Teaches(Professor2,Course3), Teaches*(*Professor3,Course4*), *Takes(Student1,Course1), Takes*(*Student2,Course1*), *Takes(Student2,Course2), Takes*(*Student3,Course2*), *Takes(Student3,Course3), Takes*(*Student4,Course2*), *Takes(Student4,Course3), Takes*(*Student4,Course4*), *Takes*(*Student5,Course4*), *Supervises*(*Professor3,Student5*)}

b. *The Extensional Reduction Model*: *R*={*Professor*^{1}, *Student*^{1}, *GradStudent*^{1}, *Course*^{1}, *Teaches*^{2}, *Takes*^{2}, *Supervises*^{2}}.

   Professor^1(w_1) = { *Professor1*, *Professor2*, *Professor3*}

   Student^1(w_1) = { *Student1*, *Student2*, *Student3*, *Student4*}

   GradStudent^1(w_1) = { *Student5*}

   Course^1(w_1) = { *Course1*, *Course2*, *Course3*, *Course4*}

   Teaches^2(w_1) = { *(Professor1,Course1), (Professor2,Course2), (Professor2,Course3), (Professor3,Course4)*}

   Takes^2(w_1) = { *(Student1,Course1), (Student2,Course1), (Student2,Course2), (Student3,Course2), (Student3,Course3), (Student4,Course2), (Student4,Course3), (Student4,Course4), (Student5,Course4)*}

   Supervises^2(w_1) = { (*Professor3,Student5*)}

The seven relationships defined above are defined, not only for the actual world, but also for all the possible worlds. The possible worlds can be defined by different arrangements between entities in the world. So another possible world, w_2 for example, would be a world that is shown in Figure 21 in which *Course2* is taught by *Professor1* and *Student4* takes only two courses instead of 3. It is also worth noting that, $Professor^1$ is a set of unary relations, while *Professor 1* is an instance (extension) that exists in the domain. And since the extensional reduction model describes concepts as a set of unary relations, the unary relation $Professor^1$ describes the concept *Professor* in the extensional reduction model. The following can be inferred from the figure:

$Professor^1(w_2)$ = { *Professor1, Professor2, Professor3*}

$Student^1(w_2)$ = { *Student1, Student2, Student3, Student4*}

$GradStudent^1(w_2)$ = { *Student5*}

$Course^1(w_2)$ = { *Course1, Course2, Course3, Course4*}

$Teaches^2(w_2)$ = { *(Professor1,Course1), (Professor1,Course2), (Professor2,Course3), (Professor3,Course4)*}

$Takes^2(w_2)$ = { *(Student1,Course1), (Student2,Course1), (Student2,Course2), (Student3,Course2), (Student3,Course3), (Student4,Course3), (Student4,Course4), (Student5,Course4)*}

$Supervises^2(w_2)$ = { (*Professor3,Student5*)}

And so on until all possible worlds are exhausted.

c. *The intensional Model*: $R = \{R^2\}$, R^2 = { *Teaches*(*Professor*,*Course*), *Takes*(*Student*,*Course*), *Supervises*(*Professor*,*GradStudent*)}

5- *The Properties*: Neither the extensional model nor the extensional reduction models capture the properties when formally treating a conceptualization. On the other hand, the intensional model treats the properties as primitives, as such, they are considered when formalizing a conceptualization. Although there are no properties shown in Figure 21, we will assume that every *Professor* is defined by

(Professor ID, Date of Birth, Name, Rank), every *Student* is defined by (student ID, Date of Birth, Name), every *GradStudent* is defined by (student ID, Staff ID, Date of Birth, Name) and every *Course* is defined by (Code, Title, Syllabus)

a. *The Extensional Model*: Does not capture properties

b. *The Extensional Reduction Model*: Does not capture properties

c. *The Intensional Model*: *P* = { *Professor.ID*, *Professor.DOB*, *Professor.Name*, *Professor.Rank*, *Student.ID*, *Student.DOB*, *Student.Name*, *GradStudent.ID*, *GradStudent.StaffID*, *GradStudent.DOB*, *GradStudent.Name*, *Course.Code*, *Course.Title*, *Course.Syllabus*}

6- *The Domain*: Both the extensional and the extensional reduction model treat the domain as the set of entities in the world. On the other hand, the intensional model treats the domain as the set of all Concept, Conceptual Relations, and intensional properties. Following is the domain based on the three formalizations:

a. *The Extensional Model*: *D* = { *Professor1*, *Professor2*, *Professor3*, *Course1*, *Course2*, *Course3*, *Course4*, *Student1*, *Student2*, *Student3*, *Student4*, *Student5*}

b. *The Extensional Reduction Model: D* = { *Professor1*, *Professor2*, *Professor3*, *Course1*, *Course2*, *Course3*, *Course4*, *Student1*, *Student2*, *Student3*, *Student4*, *Student5*}

c. *The Intensional Model: D* = { *C*, *R*, *P*}

C = {*Professor*, *Course*, *Student* , *GradStudent*}

R = {*Teaches*(*Professor*,*Course*), *Takes*(*Student*,*Course*), *Supervises*(*Professor*,*GradStudent*)}

P = { *Professor.ID*, *Professor.DOB*, *Professor.Name*, *Professor.Rank*, *Student.ID*, *Student.DOB*, *Student.Name*, *GradStudent.ID*, *GradStudent.StaffID*, *GradStudent.DOB*, *GradStudent.Name*, *Course.Code*, *Course.Title*, *Course.Syllabus*}

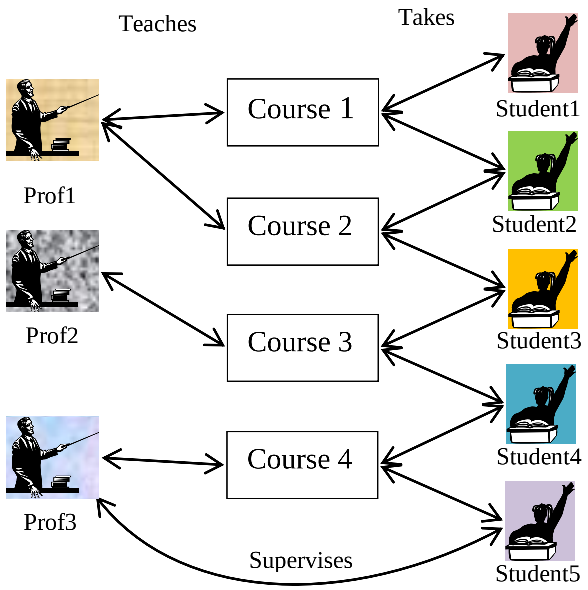

Figure 22: Another Snapshot of a University-Department World

After defining all the factors above, it is time to formally define the conceptualization according to the three models. The extensional model treats the conceptualization based on the definition of conceptualization that can be found in (Gruber 1993) and (Gruber 1995) which defines the conceptualization as “the objects, concepts, and other entities that are presumed to exist in some area of interest and the relations that hold amongst them”. The extensional reduction model, however, is based on a definition for the conceptualization that is found in (Guarino, Oberle, and Staab 2009) which defines the conceptualization as “an intensional semantic structure that encodes the implicit roles constraining the structure of a piece of reality”. And finally, the intensional model treates the conceptualization based on the definition that is found in (Xue 2010) and further refined in (Ali and Ghenniwa 2012) which defines the conceptualization to be “an abstraction that consists of the relevant concepts and relations that exist in certain domain”. The conceptualization for the world shown in Figure 22 according to the three models is:

a- *The Extensional Model*: *E = <D, F, R>*

D = { *Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

F = {}

R={ *Teaches(Professor1,Course1), Teaches*(*Professor2,Course2*), *Teaches(Professor2,Course3), Teaches*(*Professor3,Course4*), *Takes(Student1,Course1), Takes*(*Student2,Course1*), *Takes(Student2,Course2), Takes*(*Student3,Course2*), *Takes(Student3,Course3), Takes*(*Student4,Course2*), *Takes(Student4,Course3), Takes*(*Student4,Course4*), *Takes*(*Student5,Course4*), *Supervises*(*Professor3,Student5*)}

b- *The Extensional Reduction Model*: *E = <D, W, R>*

D = { *Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

$W = \{ w_1, w_2, w_3, \ldots\}$

$R=\{Professor^1, Student^1, GradStudent^1, Course^1, Teaches^2, Takes^2{}_{,} Supervises^2\}.$

c- *The Intensional Model*: *E = <D, K>*

D = { *C, R, P*}

C = {*Professor, Course, Student , GradStudent*}

R = {*Teaches*(*Professor,Course*), *Takes*(*Student,Course*), *Supervises*(*Professor,GradStudent*)}

P = { *Professor.ID*, *Professor.DOB*, *Professor.Name*, *Professor.Rank*, *Student.ID*, *Student.DOB*, *Student.Name*, *GradStudent.ID*, *GradStudent.StaffID*, *GradStudent.DOB*, *GradStudent.Name*, *Course.Code*, *Course.Title*, *Course.Syllabus*}

K: is a set of extensionalization functions that relate extensions to intensions.

As demonstrated earlier, an ontology specifies a conceptualization. From an engineering perspective, there can be several ontologies that specify the same conceptualization based on the need. The domain expert, the designer who designs the ontology, and even the application it is serving can be a factors in making one ontology different from another. Before we examine the ontological commitments and the ontologies and apply it to data integration, we will analyze the formal treatment of conceptualization in the three models. In order to make it clear, another snapshot of the world will be presented in Figure 23.

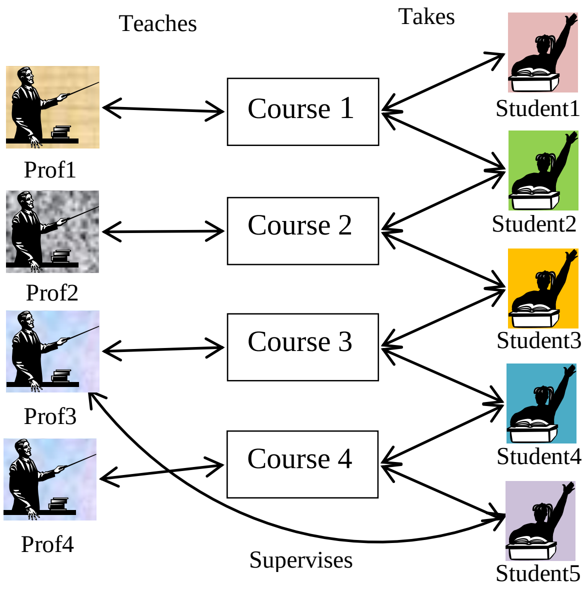

Figure 23: A Third Snapshot of a University-Department World

As can be seen in Figure 23, another *Professor* is introduced that is now teaching *Course4*. We will have a look at how the three models react to this change in the following sections.

5.2 Extensional treatment of conceptualization

Consider the snapshot of the world shown in Figure 21. To the extensional model, the conceptualization *E* is described in terms of the domain *D*, the set of functions *F* on the domain entities, and the relationships *R* between the domain entities. The set of functions in the current snapshot of the world happens to be empty. In other snapshots of the worlds or in other domains of interest it may have values. As such the extensional reduction model describes the conceptualization based on the snapshot of the world as follows:

$E_{e1} = (D_{e1}, F_{e1}, R_{e1})$ where,

D_{e1} = {*Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

$F_{e1} = \{\}$

R_{e1}={ *Teaches(Professor1,Course1), Teaches*(*Professor2,Course2*), *Teaches(Professor2,Course3), Teaches*(*Professor3,Course4*), *Takes(Student1,Course1), Takes*(*Student2,Course1*), *Takes(Student2,Course2), Takes*(*Student3,Course2*), *Takes(Student3,Course3), Takes*(*Student4,Course2*), *Takes(Student4,Course3), Takes*(*Student4,Course4*), *Takes*(*Student5,Course4*), *Supervises*(*Professor3,Student5*)}

Now, let us consider another snapshot of the world shown in Figure 22. The extensional description of the conceptualization based on the snapshot of the world in Figure 22 is as follows:

$E_{e2} = (D_{e2}, F_{e2}, R_{e2})$ where,

D_{e2} = {*Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

F_{e2} = {}

R_{e2}={ *Teaches(Professor1,Course1), Teaches(Professor1,Course2), Teaches(Professor2,Course3), Teaches(Professor3,Course4), Takes(Student1,Course1), Takes(Student2,Course1), Takes(Student2,Course2), Takes(Student3,Course2), Takes(Student3,Course3), Takes(Student4,Course3), Takes(Student4,Course4), Takes(Student5,Course4), Supervises(Professor3,Student5)*}

It is clear that $R_{e1} \neq R_{e2}$ and as such, $E_{e1} \neq E_{e2}$. According to (Guarino and Giaretta 1995), “this is what originates the troubles”. This is because the conceptualization is about concepts and should not change if the state of the world changes (Guarino and Giaretta 1995). In order to complete our argument, the snapshot of the world in Figure 23 will also be examined here. The main difference in the snapshot of the world in Figure 23 is the addition of a new professor *Professor4*. This is described in the extensional model as:

$E_{e3} = (D_{e3}, F_{e3}, R_{e3})$ where,

D_{e3} = { *Professor1, Professor2, Professor3, Professor4, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

F_{e3} = {}

R_{e3}={ *Teaches(Professor1,Course1), Teaches(Professor2,Course2), Teaches(Professor3,Course3), Teaches(Professor4,Course4), Takes(Student1,Course1), Takes(Student2,Course1), Takes(Student2,Course2), Takes(Student3,Course2), Takes(Student3,Course3), Takes(Student4,Course3), Takes(Student4,Course4), Takes(Student5,Course4), Supervises(Professor3,Student5)*}

It can be noticed from the formulations above that $D_{e2} \neq D_{e3}$ and $R_{e2} \neq R_{e3}$ and as such, $E_{e2} \neq E_{e3}$. This means, the introduction of an instance, of a class that is already captured in the conceptualization, changes the conceptualization. And that is why the extensional model is good for describing a static snapshot of the system but does not fulfill the requirement of a dynamic open environment.

5.3 Extensional Reduction treatment for conceptualization

As mentioned earlier, according to the authors in (Guarino and Giaretta 1995), changing the conceptualization without the meanings change is what originates their troubles. This is because the conceptualization is about concepts and should not change if the state of the world changes. The extensional reduction model based on the possible world approach tried to overcome some of the limitations of the extensional model. According to the extensional reduction model, the conceptualization for the snapshot of the world in Figure 21 is described as follows:

$E_{er1} = (D_{er1}, W_{er1}, R_{er1})$ where,

D_{er1} = {*Professor1, Professor2, Professor3, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

$W_{er1} = \{w_1, w_2, w_3, \ldots w_n\}$ where n is the number of all possible arrangement of the entities in the domain of interest.

R_{er1}={*Professor*^{1}, *Student*^{1}, *GradStudent*^{1}, *Course*^{1}, *Teaches*^{2}, *Takes*^{2}, *Supervises*^{2}}

Professor$^1(w_1)$ = { *Professor1, Professor2, Professor3*}

Student$^1(w_1)$ = { *Student1, Student2, Student3, Student4*}

GradStudent$^1(w_1)$ = { *Student5*}

Course$^1(w_1)$ = { *Course1, Course2, Course3, Course4*}

Teaches$^2(w_1)$ = { *(Professor1,Course1), (Professor2,Course2), (Professor2,Course3), (Professor3,Course4)*}

Takes$^2(w_1)$ = { *(Student1,Course1), (Student2,Course1), (Student2,Course2), (Student3,Course2), (Student3,Course3), (Student4,Course2), (Student4,Course3), (Student4,Course4), (Student5,Course4)*}

Supervises$^2(w_1)$ = { (*Professor3,Student5*)}

Assuming w_2 is the world in which *Professor1* teaches *Course2*, and *Student4* does not take *Course2*, the following can be described:

$Professor^1(w_2)$ = { *Professor1, Professor2, Professor3*}

$Student^1(w_2)$ = { *Student1, Student2, Student3, Student4*}

$GradStudent^1(w_2)$ = { *Student5*}

$Course^1(w_2)$ = { *Course1, Course2, Course3, Course4*}

$Teaches^2(w_2)$ = { *(Professor1,Course1), (Professor1,Course2), (Professor2,Course3), (Professor3,Course4)*}

$Takes^2(w_2)$ = { *(Student1,Course1), (Student2,Course1), (Student2,Course2), (Student3,Course2), (Student3,Course3), (Student4,Course3), (Student4,Course4), (Student5,Course4)*}

$Supervises^2(w_2)$ = { (*Professor3,Student5*)}

And so on until all different possible arrangements between the entities in the world are exhausted.

Now, consider the second snapshot of the world shown in Figure 22. As mentioned earlier, the relations in all possible worlds are captured by the conceptualization, E_{er1}. As such, the description of the conceptualization E_{er2} based on Figure 22 will be equivalent to the description of the conceptualization E_{er1}, derived from Figure 21. We can say that, $D_{er1}= D_{er2}$, $W_{er1}= W_{er2}$, and $R_{er1} = R_{er2}$ and so, $E_{er1} = E_{er1}$. This is an advantage over the extensional model. However, when we describe the conceptualization according to the extensional reduction model based on the snapshot of the world shown in Figure 23 we get the following:

$E_{er3} = (D_{er3}, W_{er3}, R_{er3})$ where,

D_{er3} = {*Professor1, Professor2, Professor3, Professor4, Course1, Course2, Course3, Course4, Student1, Student2, Student3, Student4, Student5*}

W_{er3} = {w_1, w_2, w_3, ...w_m} where *m* is the number of all possible arrangement of the entities in the domain of interest.

R_{er3}={$Professor^1$, $Student^1$, $GradStudent^1$, $Course^1$, $Teaches^2$, $Takes^2$, $Supervises^2$}

$Professor^1(w_1)$ = { *Professor1, Professor2, Professor3, Professor4*}

$Student^1(w_1)$ = { *Student1, Student2, Student3, Student4*}

$GradStudent^1(w_1)$ = { *Student5*}

$Course^1(w_1)$ = { *Course1, Course2, Course3, Course4*}

$Teaches^2(w_1)$ = { *(Professor1,Course1), (Professor2,Course2), (Professor3,Course3), (Professor4,Course4)*}

$Takes^2(w_1)$ = { *(Student1,Course1), (Student2,Course1), (Student2,Course2), (Student3,Course2), (Student3,Course3), (Student4,Course3), (Student4,Course4), (Student5,Course4)*}

$Supervises^2(w_1)$ = { (*Professor3,Student5*)}

As can be seen from the above formulations, when an instance *Professot4* is added to the world, the domain changes, the set of possible world changes, and the set of relations also changes. This can be seen when looking at the relationship *$Professor^1$* for R_{er3}, the domain D_{er3}, and the set of possible worlds W_{er3}. It can be noticed that $D_{er2} \neq D_{e3}$, $R_{er2} \neq R_{er3}$, and $R_{er2} \neq R_{er3}$, and as such, $E_{er2} \neq E_{er3}$. Even though what was introduced to the system is not a new concept, we can see that the conceptualization changes. Then we can infer that, when using the extensional reduction description of the conceptualization, the conceptualization will change even though the meanings do not change. That is what caused the extensional reduction system to be appropriate for a system with a fixed set of entities, but not for a dynamic system in open environment where entities can enter and leave the system at any time.

5.4 Intensional treatment for conceptualization

According to the intensional model, the concepts, relations, and properties are considered as primitives, and as such, irreducible. As such, it captures the concepts, conceptual relations, and intensional properties in a given domain of interest. For a snapshot of the world, the intensional model describes the conceptualization intensionally according to the theory of PRP (Bealer 1979). It treats the concepts, relations, and properties, as primitive, and as such, irreducible. Following is the, intensional, description of the conceptualization derived from the snapshot of the world in Figure 21:

$E_{i1} = (D_{i1}, \mathcal{K}_{\mathrm{i1}})$

$D_{i1} = (C_{i1}, R_{i1}, P_{i1})$

$C_{i1} = \{Professor, Course, Student ,GradStudent\}$

$R_{i1} = \{R^2\}$

$R^2 = \{Teaches(Professor,Course), Takes(Student,Course),$
$Supervises(Professor,GradStudent)\}$

P_{i1} = {*Professor.ID*, *Professor.DOB*, *Professor.Name*, *Professor.Rank*, *Student.ID*, *Student.DOB*, *Student.Name*, *GradStudent.ID*, *GradStudent.StaffID*, *GradStudent.DOB*, *GradStudent.Name*, *Course.Code*, *Course.Title*, *Course.Syllabus*}

As can be seen in the above formulations, the intensional model captures the classes of objects as concepts and captures their properties and the conceptual relations between them.

Now looking at the snapshot of the world in Figure 22, following is the intensional description of the conceptualization that can be derived from Figure 22:

$E_{i2} = (D_{i2}, \mathcal{K}_{\mathrm{i2}})$

$D_{i2} = (C_{i2}, R_{i2}, P_{i2})$

C_{i2} = {*Professor*, *Course*, *Student* ,*GradStudent*}

R_{i2} = {R^2}

R^2 = {*Teaches*(*Professor*,*Course*), *Takes*(*Student*,*Course*), *Supervises*(*Professor*,*GradStudent*)}

P_{i2} = {*Professor.ID*, *Professor.DOB*, *Professor.Name*, *Professor.Rank*, *Student.ID*, *Student.DOB*, *Student.Name*, *GradStudent.ID*, *GradStudent.StaffID*, *GradStudent.DOB*, *GradStudent.Name*, *Course.Code*, *Course.Title*, *Course.Syllabus*}

One can see that $C_{i1} = C_{i2}$, $R_{i1} = R_{i2}$, and $P_{i1} = P_{i2}$. As such $E_{i1} = E_{i2}$, which means that the conceptualization between the snapshots in both Figure 21 and Figure 22 did not change. This is because the meaning did not change. All that is changed between the two snapshots of the world is an instance, or an extension, of a relationship that is already captured in the conceptualization.

Now looking at the snapshot of the world described by Figure 23, the intensional description of the conceptualization is as follows:

$E_{i3} = (D_{i3}, \mathcal{K}_{\text{i3}})$

$D_{i3} = (C_{i3}, R_{i3}, P_{i3})$

C_{i3} = {*Professor*, *Course*, *Student* ,*GradStudent*}

R_{i3} = {R^2}

R^2 = {*Teaches*(*Professor*,*Course*), *Takes*(*Student*,*Course*), *Supervises*(*Professor*,*GradStudent*)}

P_{i3} = {*Professor.ID*, *Professor.DOB*, *Professor.Name*, *Professor.Rank*, *Student.ID*, *Student.DOB*, *Student.Name*, *GradStudent.ID*, *GradStudent.StaffID*, *GradStudent.DOB*, *GradStudent.Name*, *Course.Code*, *Course.Title*, *Course.Syllabus*}

As can be noticed, $C_{i1} = C_{i2} = C_{i3}$, $R_{i1} = R_{i2} = R_{i3}$, and $P_{i1} = P_{i2} = P_{i3}$. As such $E_{i1} = E_{i2} = E_{i2}$. This is because the meaning did not change between the three snapshots of the world. All that is changed in Figure 23 is an instance, extensionalization, of a concept that is already captured in the conceptualization. As such, we should not expect the conceptualization to change. This shows how the intensional model is capable of addressing the dynamic nature of open environment and how it adapts to changes in the environment as long as meanings do not change.

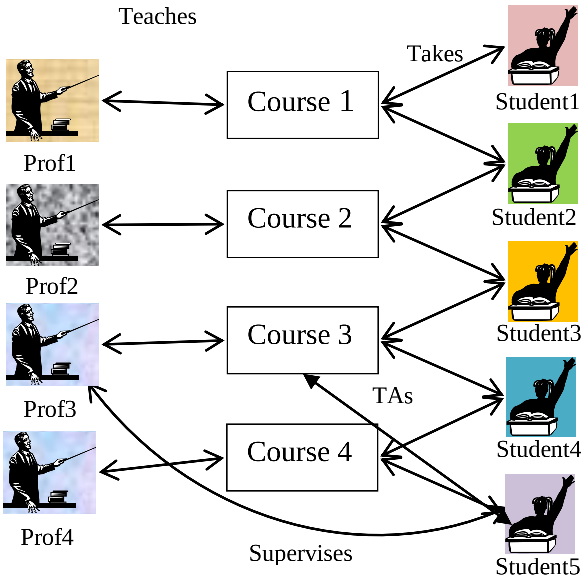

Figure 24: The Introduction of an Instance of a New Relation

For the sake of further illustration and to complete of the argument, we will show when a change in the world is expecting to change the conceptualization. At which point, the change in the conceptualization is justified. Let us take a look at the snapshot of the world shown in Figure 24. In Figure 24 a new relationship is added, which now makes the student *Student5* a TA for the course *Course3*. Below is the intensional description of the conceptualization derived from Figure 24:

$E_{i4} = (D_{i4}, \mathcal{K}_{i4})$

$D_{i4} = (C_{i4}, R_{i4}, P_{i4})$

$C_{i4} = \{Professor, Course, Student\ ,GradStudent\}$

$R_{i4} = \{R^2\}$

$R^2 = \{Teaches(Professor,Course), Takes(Student,Course),$
$Supervises(Professor,GradStudent), TAs(GradStudent, Course)\}$

$P_{i4} = \{Professor.ID, Professor.DOB, Professor.Name, Professor.Rank, Student.ID,$
$Student.DOB, Student.Name, GradStudent.ID, GradStudent.StaffID, GradStudent.DOB,$
$GradStudent.Name, Course.Code, Course.Title, Course.Syllabus\}$

It is obvious from the above formulations that $R_{i4} \neq R_{i3}$, and as such $E_{i4} \neq E_{i3}$. And this is understandable because what was introduced to the system is not just an instance of a concept, a property, or a conceptual relation that is captured by the conceptualization. Rather, an instance of a new conceptual relation called, TAs, is introduced. As such, the meanings have changed and this justifies the changes in the conceptualization, if this change is deemed relevant to our conceptualization. On the other hand, any change that is introduced to the environment and is not deemed relevant to the conceptualization is going to be abstracted out and the conceptualization will remain unchanged.

5.5 Comparing the three Models

Table 2 below compares the main aspects of the three models; the extensional model, the extensional reduction model, and the intensional model. The items in Table 2 will sum up the main differences we have discussed so far.

Table 2: Comparison of the Main Aspects of the Three Models for Describing a Conceptualization

Model	Extensional	Extensional Reduction	Intensional
Entities	Captures entities	Captures entities	Does not capture entities
Concepts (C)	Does not capture concepts	Captures concepts as unary relations	Captures Concepts
Possible Worlds (W)	No possible worlds	Every possible arrangement of entities is represented by a possible world	No possible worlds
Relations (R)	Ordinary extensional relations	Captures the extensional relations in all possible worlds and captures the classes of objects as unary relations	Captures conceptual intensional relations
Properties (P)	Does not capture properties	Does not capture properties	Captures the properties of the concepts in the domain of interest
Domain (D)	The set of entities	The set of entities	The set of concepts, conceptual relations, and properties
Conceptualization $\boldsymbol{C}$	Changes if any relationship between the domain entities changes	Changes with the addition of a new entity even if it is an instance of a class that is already captured	Changes only if a new concept, conceptual relation, conceptual property is introduced

After examining the description of the conceptualization in the three different models we will examine describe the ontology in the intensional model. As has been mentioned earlier, from an engineering perspective, an ontology specifies a conceptualization and commits to the conceptualization through an ontological commitment. There can be various views of the ontology based on several factors. These factors can include, the application for which the ontology is being designed, the domain expert, the granularity requirement, and the designer who builds the ontology. As such, this creates the heterogeneity when integrating several information systems that are built for different purposes or by different agents. Even if the data integration systems share the same domain, they can still be different. As such, some sort of mapping is required in order to integrate data residing at different data sources. Here we will consider one view of the ontology, and then other ontologies will be created for the same domain. It will, then, be shown how the various ontologies can map to one another. Figure 25 shows an ontology that specifies the conceptualization E_{i1} described in the intensional model.

$$\forall x\ \mathit{Human}(x) \rightarrow \mathit{Thing}(x)$$

$$\forall x\ \mathit{Professor}(x) \rightarrow \mathit{Human}(x)$$

$$\forall x\ ([\mathit{HasProfID}(x)]_{x} \wedge [\mathit{HasDOB}(x)]_{x} \wedge [\mathit{HasName}(x)]_{x} \wedge [\mathit{HasRank}(x))]_{x} \rightarrow \mathit{Professor}(x)$$

$$\forall x\ \mathit{Student}(x) \rightarrow \mathit{Human}(x)$$

$$\forall x\ \mathit{UndergradStudent}(x) \rightarrow \mathit{Student}(x)$$

$$\forall x\ ([\mathit{HasStudentID}(x)]_{x} \wedge [\mathit{HasDOB}(x)]_{x} \wedge [\mathit{HasName}(x))]_{x} \rightarrow \mathit{UndergradStudent}(x)$$

$$\forall x\ \mathit{GradStudent}(x) \rightarrow \mathit{Student}(x)$$

$$\forall x\ ([\mathit{HasStudentID}(x)]_{x} \wedge [\mathit{HasStaffID}(x)]_{x} \wedge [\mathit{HasDOB}(x)]_{x} \wedge [\mathit{HasName}(x))]_{x} \rightarrow \mathit{GradStudent}(x)$$

$$\forall x\ \mathit{Course}(x) \rightarrow \mathit{Thing}(x)$$

$$\forall x\ ([\mathit{HasCode}(x)]_{x} \wedge [\mathit{HasTitle}(x)]_{x} \wedge [\mathit{HasSyllabus}(x)]_{x}) \rightarrow \mathit{Course}(x)$$

$$\forall x \forall y\ ([\mathit{Teaches}(x,y)]_{x,y}) \rightarrow \mathit{Professor}(x) \wedge \mathit{Course}(y)$$

$$\forall x \forall y\ ([\mathit{Takes}(x,y)]_{x,y}) \rightarrow \mathit{Student}(x) \wedge \mathit{Course}(y)$$

$$\forall x \forall y\ ([\mathit{Supervises}(x,y)]_{x,y}) \rightarrow \mathit{Professor}(x) \wedge \mathit{GradStudent}(y)$$

$$\forall x \forall y \forall z\ ([\mathit{ReportsTo}(x,y,z)]_{x,y,z}) \leftrightarrow \mathit{Student}(x) \wedge \mathit{Professor}(y) \wedge \mathit{Course}(z) \wedge [\mathit{Takes}(x,z)]_{x,z} \wedge [\mathit{Teaches}(y,z)]_{y,z}$$

Figure 25: An Ontology for the World Shown in Figure 21

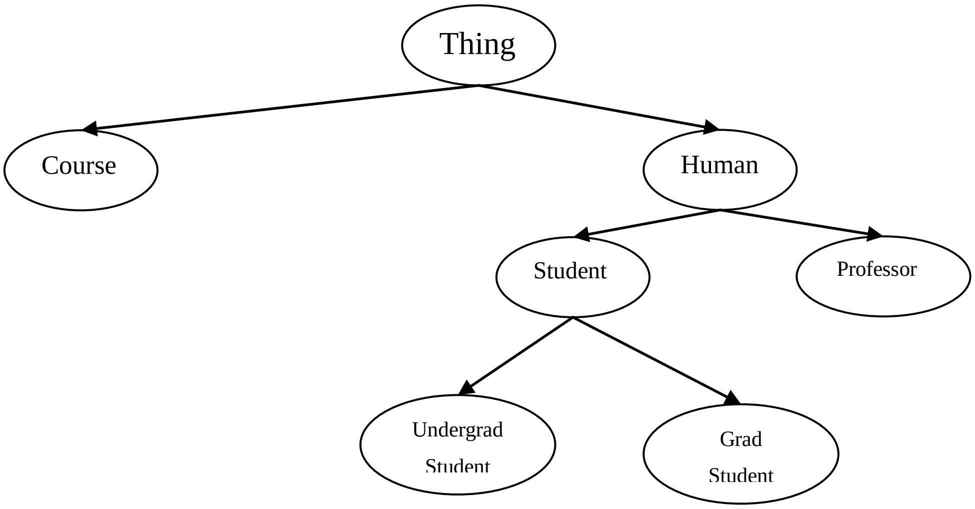

Figure 26: Taxonomical Structure for the Ontology in Figure 25

The taxonomical structure for the ontology shown in Figure 25 is displayed in Figure 26. As can be seen in Figure 26 every Human is a Thing and every Course is also a Thing. Also, every Professor is a Human and every Student is a Human. It is also shown that every Undergrad Student is a Student and every Grad Student is a Student. But this is only one possibility of so many possible ontologies that can specify the conceptualization E_{i1} above. Other applications may require different ontologies, other domain experts may have different views, and other designers may have different philosophies and experience. The following figures, Figure 27, Figure 28, and Figure 29, show three other possible ontologies that specify the same conceptualization E_{i1}. We will consider these three ontologies and the ontology shown in Figure 26 to be the ontologies for four different information systems in open environment. Ultimately, we would like to integrate the four different information systems. For this task to be achieved, the intensional model, which is described in Chapter 4, will be employed.

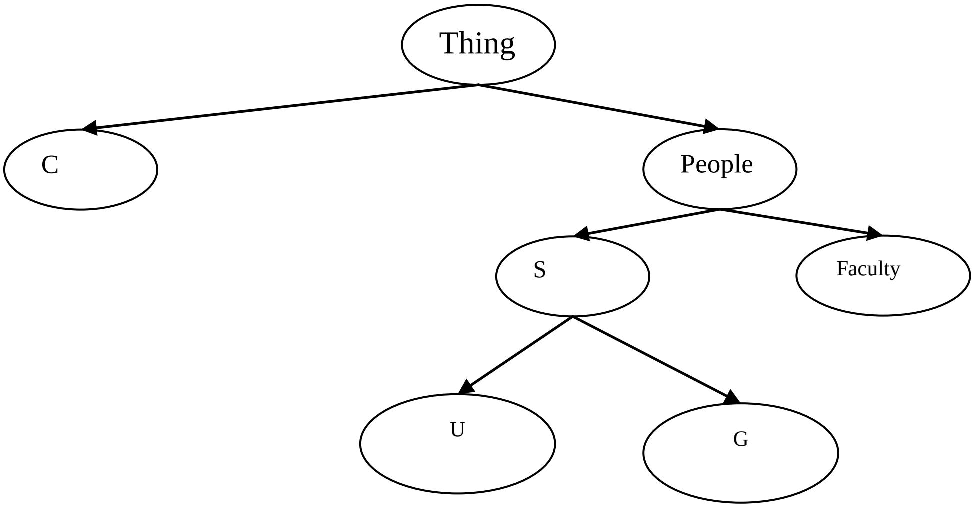

Figure 27: A Possible Ontology for Specifying the Conceptualization *E_{i1}*

5.6 Intensional Data Integration in Open Environment

Let us assume that we have several ontologies created for the same domain. All the ontologies will share the same domain of discourse but can have different designs and vocabularies. This creates a heterogeneous environment. In such environment, a query posed to the ontology of one information system can have possible answers in all other information systems accessible from this information system. In order to achieve a certain degree of autonomy, each information system will typically have a private ontology *OP*, and a global ontology *OG*. The private ontology *OP* is the full ontology for the information system, whereas, the global ontology represent what the information system chooses to share with other information systems. For illustration purposes, we will consider that the two ontologies are merged into one ontology *O* in this example. Also, according to the Mediated P2P architecture, each data source will be connected to a mediator peer and there will be local mappings between the private ontology of the mediator peer and the ontology of the data source. For simplicity, we will assume that all the data sources connected to each mediator peer are federated into one data source which shares the same ontology with the mediated peer. As such, the local mapping between the mediator peer and the local data source can be dropped for the purpose of this illustration.

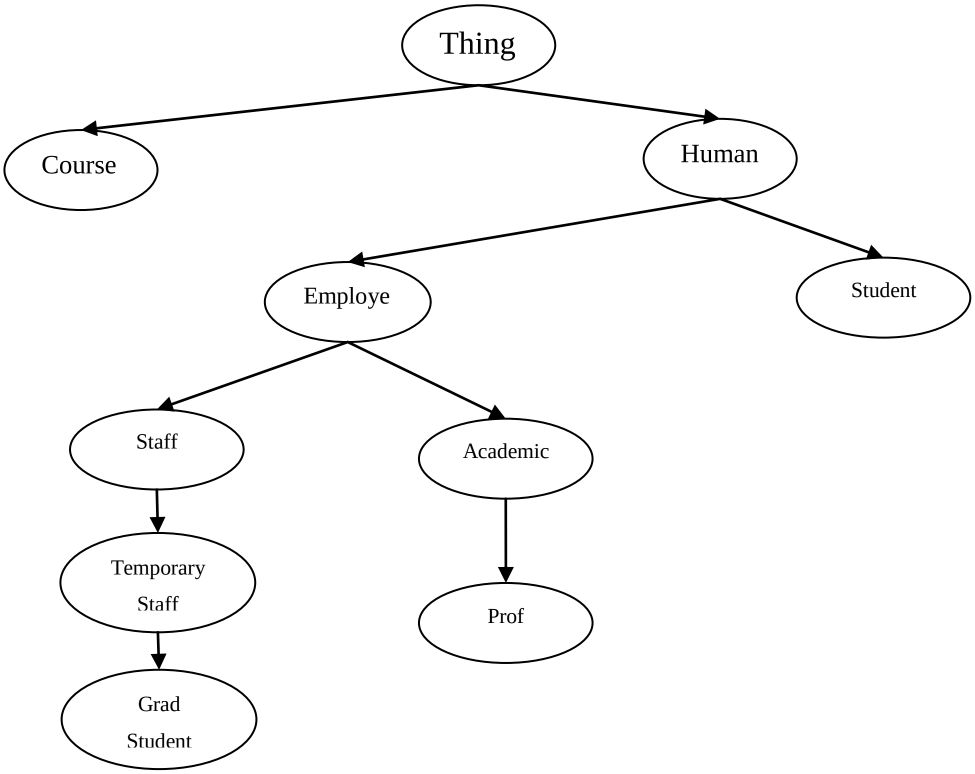

Figure 28: A Possible Ontology for Specifying the Conceptualization E_{i1}

The example that we will present is shown in Figure 31 and consist of five mediator peers, P_1, P_2, P_3, P_4, and P_5 with ontologies O_1, O_2, O_3, O_4, and O_5 respectively. The taxonomical structures of ontologies O_1, O_2, O_3, O_4, and O_5 are shown in Figure 26, Figure 27, Figure 28, Figure 29, and Figure 30 respectively. These ontologies are created for the same domain, the simplified university-department domain, but for different applications and by different agents. As can be noticed, the five ontologies are heterogeneous. In order to integrate the information the various information systems the first step is to find a mapping between the various ontologies. The global mappings only exist between the ontologies of peers that are neighbors in the network.

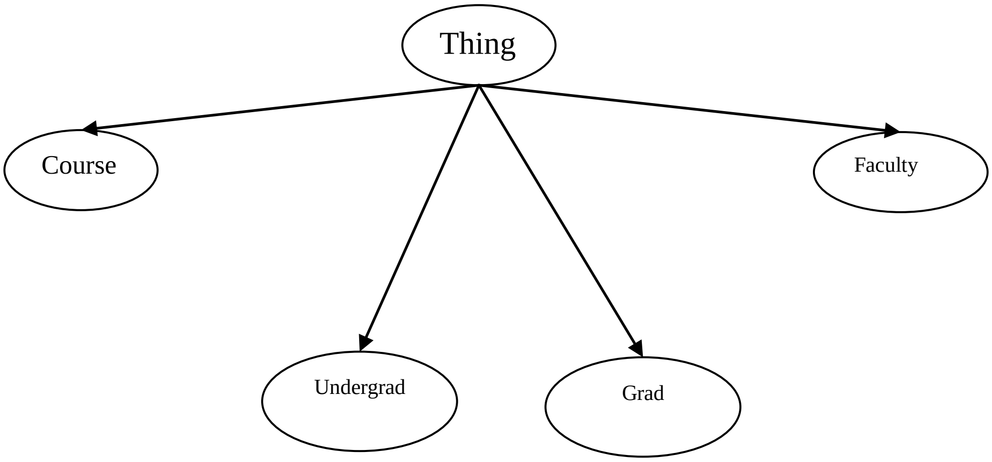

Figure 29: A Possible Ontology for Specifying the Conceptualization *E_{i1}*

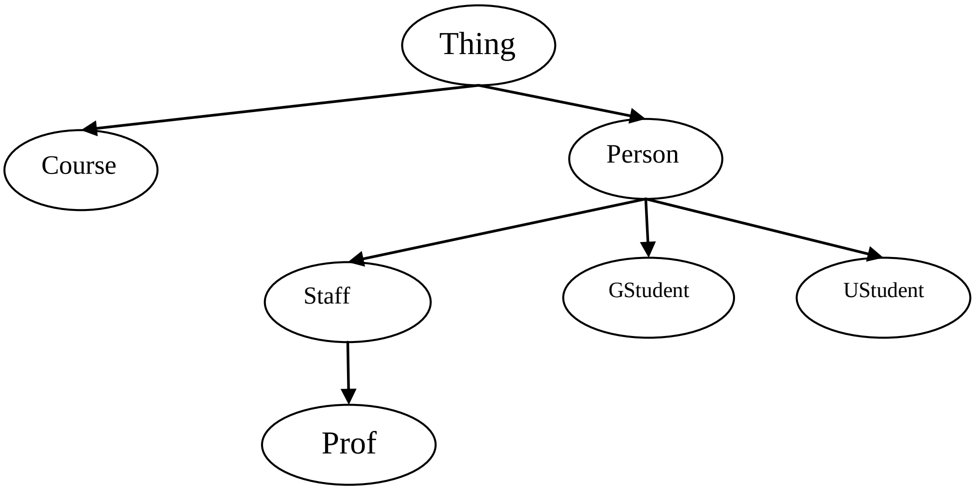

Figure 30: A Possible Ontology for Specifying the Conceptualization *E_{i1}*

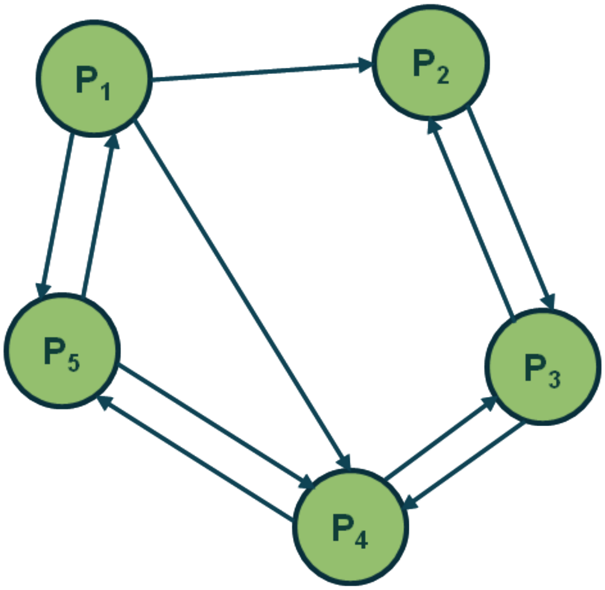

Figure 31: A Mediated P2P Network with five Mediator Peers

5.6.1 Ontology Mapping

According to the intensional model for data integration in open environment, the access between a mediator peer P_i and other mediator peers in the network is captured in the set of accessibility relations R_i. Before the set of mappings Gi from peer P_i to other peers is specified, the various sets of accessibility relations will be listed here first. Because we have five mediator peers, and because the accessibility relation can happen in one of two directions, we will have five different set of accessibility relations as follows:

$R_1 = \{(P_1, P_2), (P_1, P_4), (P_1, P_5)\}$

$R_2 = \{(P_2, P_3)\}$

$R_3 = \{(P_3, P_2), (P_3, P_4)\}$

$R_4 = \{(P_4, P_3), (P_4, P_5)\}$

$R_5 = \{(P_5, P_4), (P_5, P_1)\}$

The global mappings G_i from peer P_i to neighboring peers is based on the existence of an accessibility relation between the two peers. Considering the sets of accessibility relations described above, the global mappings for each mediated peer are as follows:

$G_1 = \{G_{12}, G_{14}, G_{15}\}$

$G_2 = \{G_{23}\}$

$G_3 = \{G_{32}\ G_{34}\}$

$G_4 = \{G_{43}, G_{45}\}$

$G_5 = \{G_{54}, G_{51}\}$

We will consider that the ontologies of the neighboring peers are used as inputs to some ontology matching algorithm, like the S-Match discussed earlier, and the results are returned. These results will determine the mapping functions between the various ontologies. We will consider that the mapping results are captured in the following mapping tables:

Table 3: The Mapping Function G_{12}

Human	*People*
Professor	*Faculty*
Student	*S*
UndergradStudent	*U*
GradStudent	*G*
Course	*C*

Table 4: The Mapping Function G_{14}

Professor	*Faculty*
UndergradStudent	*UnderGrad*
GradStudent	*Grad*
Course	*Course*

Table 5: The Mapping Function G_{15}

Human	*Person*
Professor	*Prof*
UndergradStudent	*UStudent*
GradStudent	*GStudent*
Course	*Course*

Table 6: The Mapping Function G_{23}

People	*Human*
Faculty	*Prof*
U	*Student*
G	*GradStudent*
C	*Course*

Table 7: The Mapping Function G_{32}

Human	*People*
Prof	*Faculty*
Student	*U*
GradStudent	*G*
Course	*C*

Table 8: The Mapping Function G_{34}

Prof	*Faculty*
Student	*Undergrad*
GradStudent	*Grad*
Course	*Course*

Table 9: The Mapping Function G_{43}

Faculty	*Prof*
Undergrad	*Student*
Grad	*GradStudent*
Course	*Course*

Table 10: The Mapping Function G_{45}

Faculty	*Prof*
Undergrad	*UStudent*
Grad	*GStudent*
Course	*Course*

Table 11: The Mapping Function G_{54}

Prof	*Faculty*
UStudent	*Undergrad*
GStudent	*Grad*
Course	*Course*

Table 12: The Mapping Function G_{51}

Person	*Human*
Prof	*Professor*
UStudent	*UndergradStudent*
GStudent	*GradStudent*
Course	*Course*

The mapping functions shown in tables Table 3 through Table 12 are then used to map P2P queries to one another.

5.6.2 Query Transformation

Consider a query $q_1(x) = \mathit{Professor}(x) \vee \mathit{GradStudent}(x)$ posed to the private ontology of the mediator peer P_1 in Figure 31. It's obvious that the network in Figure 31 is represented by a cyclic graph. In order to calculate all the possible answers for the query, the acyclic tree rooted at P_1 will be created. The tree can be seen in Figure 32. With the help of the mapping, the acyclic tree is then used to calculate all the possible answers. Formally speaking, there will be five distinguished global intensional epistemic logic theories. Each theory is concerning one mediator peer and its immediate network. We will refer to the global theories for P_1, P_2, P_3, P_4, and P_5 as T_{GP1}, T_{GP2}, T_{GP3}, T_{GP4}, and T_{GP5} respectively. The set of agents for each theory, as described in equation (31), are as follows:

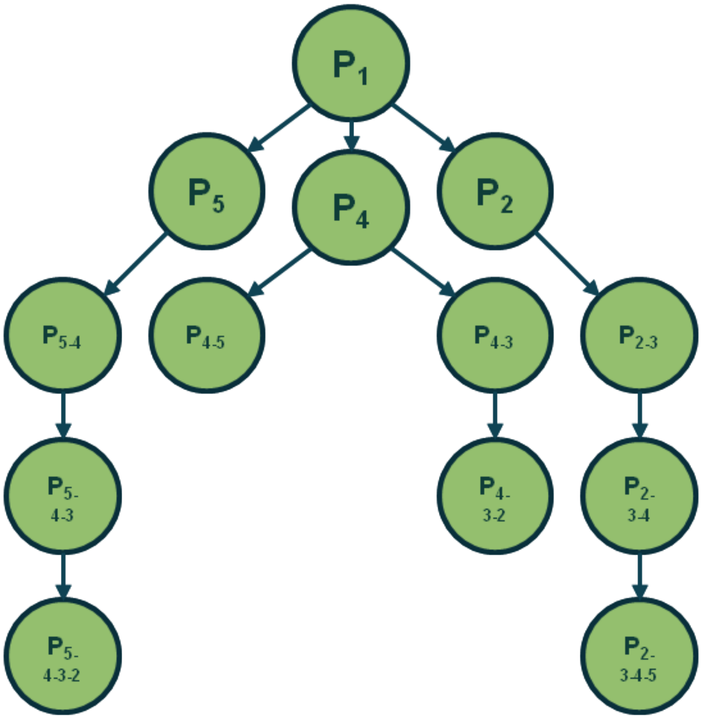
P_1
P_5
P_4
P_2
$P_{5\text{-}4}$
$P_{4\text{-}5}$
$P_{4\text{-}3}$
$P_{2\text{-}3}$
$P_{5\text{-}4\text{-}3}$
$P_{4\text{-}3\text{-}2}$
$P_{2\text{-}3\text{-}4}$
$P_{5\text{-}4\text{-}3\text{-}2}$
$P_{2\text{-}3\text{-}4\text{-}5}$

Figure 32: Acyclic Tree rooted at P_1 for the network shown in Figure 31

$AGTS_1= \{P_1, P_2, P_4, P_5\}$

$AGTS_2= \{P_2, P_3\}$

$AGTS_3= \{P_3, P_2, P_4\}$

$AGTS_4= \{P_4, P_3, P_5\}$

$AGTS_5= \{P_5, P_1, P_4\}$

The alphabets for each theory are described by equation (32). In that sense, the alphabets $\mathcal{A}_{TGP1}$ for the theory T_{GP1}, for example, is the disjoint union of the alphabets for the mediator peer P_1 and all the mediated peers accessible directly from P_1. Applying the equation to the example at hand, the following can be inferred:

$\mathcal{A}_{TGP1}$ = {P_1.*Human*, P_1.*Professor*, P_1.*Student*, P_1.*UndergradStudent*, P_1.*GradStudent*, P_1.*Course*, P_2.*People*, P_2.*Faculty*, P_2.*S*, P_2.*U*, P_2.*G*, P_2.*Course*, P_4.*Faculty*, P_4.*Undergrad*, P_4.*Grad*, P_4.*Course*, P_5.*Person*, P_5.*Staff*, P_5.*Prof*, P_5.*GStudent*, P_5.*UStudent*, P_5.*Course*}

$\mathcal{A}_{TGP2}$ = {P_2.*People*, P_2.*Faculty*, P_2.*S*, P_2.*U*, P_2.*G*, P_2.*Course*, P_3.*Human*, P_3.*Employee*, P_3.*Academic*, P_3.*Staff*, P_3.*Prof*, P_3.*TemporaryStaff*, P_3.*GradStudent*, P_3.*Course*}

$\mathcal{A}_{TGP3}$ = {P_3.*Human*, P_3.*Employee*, P_3.*Academic*, P_3.*Staff*, P_3.*Prof*, P_3.*TemporaryStaff*, P_3.*GradStudent*, P_3.*Course*, P_2.*People*, P_2.*Faculty*, P_2.*S*, P_2.*U*, P_2.*G*, P_2.*Course*, P_4.*Faculty*, P_4.*Undergrad*, P_4.*Grad*, P_4.*Course*}

$\mathcal{A}_{TGP4}$ = {P_4.*Faculty*, P_4.*Undergrad*, P_4.*Grad*, P_4.*Course*, P_1.*Human*, P_1.*Professor*, P_1.*Student*, P_1.*UndergradStudent*, P_1.*GradStudent*, P_1.*Course*, P_3.*Human*, P_3.*Employee*, P_3.*Academic*, P_3.*Staff*, P_3.*Prof*, P_3.*TemporaryStaff*, P_3.*GradStudent*, P_3.*Course*, P_5.*Person*, P_5.*Staff*, P_5.*Prof*, P_5.*GStudent*, P_5.*UStudent*, P_5.*Course*}

$\mathcal{A}_{TGP5}$ = {P_5.*Person*, P_5.*Staff*, P_5.*Prof*, P_5.*GStudent*, P_5.*UStudent*, P_5.*Course*, P_1.*Human*, P_1.*Professor*, P_1.*Student*, P_1.*UndergradStudent*, P_1.*GradStudent*, P_1.*Course*, P_4.*Faculty*, P_4.*Undergrad*, P_4.*Grad*, P_4.*Course*}

Talking about the theory T_{GP1} for mediator peer P_1, the mapping functions in Table 3, Table 4, and Table 5 will be used to generate the intensionally equivalent query over the global ontologies of mediated peers, P_2, P_4, and P_5, respectively. Given the query $q_1(x)$ above, the intensionally equivalent queries $q_{12}(x) = G_{12}(q_1(x))$, $q_{14}(x) = G_{14}(q_1(x))$, and $q_{15}(x) = G_{15}(q_1(x))$ over the global ontologies of peers P_2, P_4, and P_5 respectively are as follows:

$q_{12}(x) = Faculty\ (x) \vee G\ (x)$

$q_{14}(x) = Faculty\ (x) \vee Grad(x)$

$q_{15}(x) = Prof\ (x) \vee GStudent(x)$

Each one of the queries above should then be forwarded to the appropriate theory in order to calculate the answer. Given the tree shown in Figure 32, the answer to each of the queries above can involve both, calculating local answers, and calculating equivalent queries over the ontologies children of each node in the tree. Calculating local answers will involve use the local mappings to map the mediator's query to other queries over the data sources of the mediator's network. On the other hand, calculating the queries over the ontologies of the other children nodes will use the P2P mappings in tables Table 3 through Table 12. This process continues until all possible answers for a certain query are calculated in a nested way. This happens when all leaves of the query answering tree are revisited. Each peer will then deliver all possible answers, the local answers plus the possible answers from its children nodes, to its parent. This will continue until all the possible answers reach the root node. If we apply equations (39) to (42), this can be expressed as follows:

$Ans_g\ (q_1(x), MP_1) = B_{P1}\ q_1(x)^1 \cup Ans_p\ (q_1(x), P_1, P_1)$

$B_{P1}\ q_1(x)^1 = \cup_{j \in S1}\ \text{ƙ}_{1\mathrm{j}}(q_1(x))$

$Ans_p\ (q_1(x), P_1, P_1) = Ans_p\ (q_1(x), P_1, P_2) \cup Ans_p\ (q_1(x), P_1, P_4) \cup Ans_p\ (q_1(x), P_1, P_5)$

$Ans_p\ (q_1(x), P_1, P_2) = B_{P1}B_{P2}\, q_1(x)^2 \cup Ans_p\ (G_{12}(q_1(x)), P_2, P_3)$

$Ans_p\ (q_1(x), P_1, P_4) = B_{P1}B_{P4}q_1(x)^2 \cup Ans_p\ (G_{14}\,(q_1(x)), P_4, P_3) \cup Ans_p\ (G_{14}\,(q_1(x)), P_4, P_5)$

$Ans_p\ (q_1(x), P_1, P_5) = B_{P1}B_{P5}q_1(x)^2\ \cup Ans_p\ (G_{15}\,(q_1(x)), P_5, P_4)$

For a leaf node, the possible answers to the query are as follows:

$Ans_p\ (q_{14}(x), P_4, P_5) = B_{P4}B_{P5}\ q_{14}(x)^2 = \cup_{j \in S5}\ ƙ_{5j}(G_{45}q_{14}(x))$

It is important to note that the provided example does not capture all the details of open environment. For example, the proposed example does not illustrate how the system will behave when an information system enters/leaves the environment. Also, the heterogeneity gap between various ontologies, in the provided example, may not require complicated matching algorithm to bridge. The example is provided, however, for illustration purposes to show how the proposed model is applied to a real life example. It also helps explaining the proposed query answering semantics.

In order to capture all the details of open environment, the system will need to have information systems that are diverse enough as will be the case with a real life open environment situation. The diversity will be on both the representation and conceptual levels. This will clearly highlight the need for using explicit semantics. It will also show the importance of using a structural ontology matching algorithm. Another important aspect that needs to be demonstrated by a more comprehensive example is the dynamic nature of open environment. Dynamicity in open environment is natural, this is due to the autonomy of the information systems associated with the environment. As such, information systems can choose to enter/leave the system at any time. A more comprehensive example will illustrate the way the system behaves when an information system enters or leaves the environment. This will show how the whole system will continue to function while allowing mediators to adapt to the changes that happen in their local networks.

5.6.3 Distributed and Loosely-Coupled Nature

It is shown that the Mediated-P2P architecture is more adequate for addressing the distributed nature of open environment. The loosely coupled nature is also address through the usage of intensional equivalence. Intensional equivalence does not impose any constraints on the extensions in order for two queries to be considered equivalent. The intensional equivalence of two queries simply means that two information systems know something about the same query. But it does not dictate what they know or require them to be consistent in their knowledge. Moreover, all agents do not have to share the same beliefs about the knowledge of an information system. Their beliefs can be different depending on the rules and mappings that exist between peers. It is also important to notice that information systems in open environment possess certain degree of autonomy. As such, the beliefs of various agents about the knowledge of an information system can differ depending on what the information system decides to share with each agent. This has been addressed through the use of the relative beliefs that are supported by the intensional epistemic logic.

5.6.4 Dynamic Nature

The Dynamic nature is address through the usage of 2*N* separate theories to represent the mediated *P2P* network. As such, if a mediated peer is removed, the local network containing this mediated peer may be affected without affecting the overall behavior of the mediated P2P network. In that sense, the number of possible answer can be changed, but the overall behavior of the network will not be compromised. It has also been shown in Chapter 3 that the use of intensional model is more adequate for the dynamic nature of open environment. This is because, using the intensional model, the conceptualization does not change when an entity, extension, enters of leaves the system.

5.6.5 Comparison with Conventional Solutions

Table 13 compares several aspects of data integration frameworks in open environment. The proposed model is compared to two various frameworks in the literature. We will be comparing the proposed model to the two frameworks proposed in (Y. D. Wang 2009) and (Xue 2010).

As shown in Table 13. The proposed model is intensional in nature. This is due to the class of logic based on which the system is modeled. Both the systems presented in (Y. D. Wang 2009) and (Xue 2010) are based on an extensional reduction model. It has been illustrated in Chapter 3 that the extensional reduction model does not adequately describe a conceptualization. It has also been demonstrated that information systems, in general, and open environments, in particular, are intensional in nature. As such, the use of intensional logic and an intensional model are natural choices for data integration in open environment.

Another important aspect shown in Table 13 is the dynamic nature of open environment. Neither (Y. D. Wang 2009) nor (Xue 2010) addressed the dynamic nature of open environment. The proposed framework, however, addresses the dynamic nature of open environment through the use of an intensional logic. The proposed framework also models a mediated P2P network, which has N number of peers, with 2N IEL theories. This enables the system to continue to function while peers adapt to the changes in their local networks.

Table 13 also highlights that the proposed model uses ontologies as the source of semantics as opposed to extracting semantics from a database schema. While database schema may contain semantics, it has been illustrated that the main focus of the database schema is the structure of data. As such, the semantics in a database schema are implicit and not maintainable. On the other hand, the main focus of ontologies is the semantics. As such, ontologies provide semantics that are explicit, maintainable, and up to date. Given the heterogeneous nature of open environment, a data integration system in open environment cannot rely on database schemas as primary sources for the semantics.

It is also shown in Table 13 that the proposed model addresses the distributed nature of open environment through the use of a Mediated P2P architecture. The architecture proposed in (Xue 2010) is a mediated architecture. Mediated architecture is inherently centralized and does not address the distributed nature of open environment. (Y. D. Wang 2009), however, propose the use of a distributed architecture. It is worth mentioning that, with the absence of a centralized control, the system presented in (Y. D. Wang 2009) will require each information system to act as a DIS on its own. This is too much to expect from every single information system in open environment.

When it comes to matching between ontologies of various information systems both (Y. D. Wang 2009) and (Xue 2010) propose the use of elementary ontology matching algorithms. The proposed model however proposes the use of a structural ontology matching algorithm. It has been demonstrated that the elementary ontology matching algorithms are not expected to yield accurate results. This is because elements of ontologies inherit semantics from their parents in the taxonomical structure. Elementary matching algorithms take concepts out of their context. As such, all the semantics that concepts inherit from their parents are not utilized by elementary matching algorithm. This is the main reason why elementary matching algorithms do not yield accurate results.

Because the two frameworks proposed in (Y. D. Wang 2009) and (Xue 2010) are based on an extensional reduction model, the two models use extensional equivalence. The extensional equivalence considers two predicates to be equivalent if they share an equivalent set of parameters. This is acceptable when dealing with a system that is extensional in nature. Data integration systems in open environment, however, are intensional in nature. As such, the proposed model employs intensional equivalence. According to the proposed model; two queries are considered to be intensionally equivalent if they are expressed in terms of equivalent intensional entities.

The two frameworks proposed in (Y. D. Wang 2009) and (Xue 2010) are based on extensional reduction model. This reflects on their description of a conceptualization. It has been demonstrated in Chapter 3 that the extensional reduction model is inadequate for describing a conceptualization. This is because it reduces the intensional matters to extensional entities. The proposed model however adopts a non reductionist approach that is based on the theory of PRP (Bealer 1979). The result is an intensional description of conceptualization. The proposed description is consistent with the view in (Guarino, Oberle, and Staab 2009). According to (Guarino, Oberle, and Staab 2009), conceptualization is about meanings. As such, a conceptualization should not change unless meanings change.

And finally, it is shown in Table 13 that, the system proposed in (Y. D. Wang 2009) does not address the representation of ontology. On the other hand, the framework presented in (Xue 2010) proposes the use of a frame-based language for representing an ontology. It has been shown in section 2.2.4.2 that frame-based languages have limited expressive power and their semantics are not precisely defined. The proposed model, however, uses intensional logic to represent ontologies. Not only does the intensional logic have clear semantics, but also, intensional logic employs singular terms to express the properties of the concepts, and the relations between concepts.

Table 13: Comparison of Data Integration Framework in Open Environment

Factor	Proposed Framework	(Y. D. Wang 2009)	(Xue 2010)
Model	Intensional	Extensional Reduction	Extensional Reduction
Dynamic Nature	Addressed with the intensional model	Not addressed	Not Addressed
Source of Semantics	Ontology	Ontological View (Extensional Reduction)	D.B Schema
Architecture	Mediated P2P	Distributed (Web Services and Agents)	Mediated
Mapping	Structural and Semantic-based	Elementary	Elementary and syntactical-based
Equivalence	Intensional	Extensional	Extensional
Conceptualization	Intensional	Extensional Reduction	Extensional Reduction
Ontology Representation	Intensional Logic	N/A	Frame Language

5.7 Completeness, and Soundness

In this section we discuss the soundness and completeness of the proposed model for mediated P2P data integration based on the intensional epistemic logic. We start by defining the soundness and completeness in the context of the proposed data integration system. The soundness and completeness can be defined as follows:

Completeness: Let us consider a mediated P2P data integration system MP2P, two mediator peers MP_i and MP_j, a data source S_{jk} in the mediated network of MP_j, a domain D_j for the mediator peer MP_j, and a query $q_i(x)$ posed to the global ontology OG_i of mediator peer MP_i. If $q_{jk}(x)$ is local query at data source S_{jk} that is intensionally equivalent to the global query $q_i(x)$, and the set of tuples $c \in D_j$ is the local answer to query $q_{jk}(x)$, then the set of tuples c is part the global answer for the query $q_i(x)$.

In other words, all possible answers are included in the global answer.

Soundness: Let us consider a mediated P2P data integration system MP2P, a mediator peers MP_i, and a query $q_i(x)$ posed to the global ontology OG_i of mediator peer MP_i. If tuple c is part of the global answer for query $q_i(x)$, then either $c \in D_i$ is part of the local answer for the intensionally equivalent local query $c \in D_j$ is part of the local answer for the intensionally equivalent local query $q_{ik}(x)$ at some data source S_{ik} in the mediated network of a peer $MP_{i,}$ or $q_{jk}(x)$ at some data source S_{jk} in the mediated network of a peer MP_j that is accessible from MP_i.

In other words, all the answers in the global answer are possible answers.

The following theorem proves the soundness and completeness of the proposed mediated P2P intensional data integration model.

Theorem 1: let $MP2P = \{MP_i | 1 \leq i \leq N\}$ be a mediated P2P data integration system in open environment, S_{jk} a data source for $MP2P$ in the mediated network of mediator peer MP_j, $q_i(x)$ is a query over the private ontology OP_i of a mediator peer MP_i, MP_i has access, direct or indirect P2P connection, to mediator peer MP_j, $q_j(x)$ is the intensionally equivalent query to $q_i(x)$ over the global ontology OG_j of mediator peer MP_j after applying the proper sequence of global P2P mappings, to the query $q_i(x)$, and the arity of the tuple of variables x is n. Then, for every tuple of constants $c \in D_j$ of arity n, the following equation holds:

$$c \in Ans_g(q_i(x), P_i) \ iff \ c \in Ans_l(q_j(x), P_j, S_{jk}) \qquad \textbf{(44)}$$

Proof:

We want to prove that all the results returned as part of $Ans_g(q_i(x))$ are either local answers to the mediated network of MP_i or the local network of another peer MP_j accessible from MP_i (Soundness).

In order to prove this we are going to prove the following:

1- The algorithm in Figure 20 for calculating the acyclic graph from the cyclic graph returns paths to accessible peers only. We will prove this by induction. First, let us assume that a path that is not accessible from the original peer is returned. This will only happen if a child y is added to a parent node x such that y$\notin R(x)$. However, step (9) in the algorithm allows the creation of a child only if there is accessibility relation between the child and the parent. And so, it is impossible to have a path in the acyclic graph that is not accessible in the original graph. As such, the algorithm in Figure 20 returns paths to accessible peers only □.

2- If a query transformation is possible, there is both accessibility relationship between the peers and a mapping between the two queries exists. Let us assume that a query $q1(x)$ over mediated peer MP_i is transformable to a query $q_2(x)$ over a mediated peer MP_j. This implies that $q_2(x) = G_{nj}G_{(n-1)n}\ldots G_{i1}(q_1(x))$ which means, there must be a series of n mapping between $q_1(x)$ and $q_2(x)$ in order for the query to be transformable. If the mapping does not exist, the transformation of the query will be impossible □.

Also, each peer is formalized by a global theory and a local theory. The global theory for peer MP_i is T_{GPi} with a set of agents $AGTS$ specified in equation (31) as

$AGTS = \{P_i\} \cup \{P_j | MP_j \in R_i(MP_i)\}$ through which the transformation of the query is made possible. The equation above and equations (32), (33), and (34) show that, the transformation of a query implying the accessibility relation, is the basis for each global theory formalizing a mediated peer. This means that, the transformation will only be possible when there is accessibility relation. By definition, the accessibility relationship is transitive. As such, if the query $q_1(x)$ is transformable to query $q_2(x)$ this implies there is an accessibility relationship between the two peers □.

3- If a query $q_1(x)$ over a mediated peer MP_i is transformable to a local query $q_2(x)$ over a data source S_{ik}, then the corresponding data source belongs to the local network of the mediated peer and a local mapping exists: From equations (40) and (43) the local answer at source S_{ik} is part of the global answer at a mediated peer MP_i only if the data source belongs to the local network of mediated peer MP_i. Also, each peer MP_i is formalized by two theories; a global theory T_{GPi} and a local theory T_{LPi}. The local theory represents each mediated peer's local network. From equations (37) and (38), the there are assertions in each local theory for the local mappings between the mediated peer and the data sources. If a mapping between the mediated peer and a data source does not exist, the assertion does not exist. As such the local transformation of the query is not possible. And so, if the query transformation is achievable, this implies that the data source is part of the peer's local network and that a mapping exists □.

We also want to prove that, if a tuple of constants c is part of the answers returned from the local query at a data source in the local network of mediated peer MP_i, or the mediated network of a mediated peer MP_j accessible from MP_i; then c is part of the global answer $Ans_g(q_i(x))$ (completeness).

In order to prove that, the following needs to be proven:

1- The algorithm in Figure 20 for calculating the query answering tree returns all possible paths, from the peer to which the query is posed, to all the peers accessible from the root peer: This will be proven by induction. First, let us assume that a path that is accessible from the original peer is not returned. This will only happen if a child y is not added to a parent node x such that y∈$R(x)$. However, step (9) in the algorithm allows the creation of a child for all the labels accessible from the current mediated peer. And, since R is transitive relation by definition, then if an accessibility relation exists between a mediated peer MP_i and MP_j, and an accessibility relationship exists between mediated peers MP_j and MP_k, then an accessibility relation exists between MP_i and MP_k. And so, it is impossible for a path that is accessible in the original graph to be missed in the acyclic graph. And so, the algorithm in Figure 20 returns all possible paths to accessible mediated peers □.

2- If a mapping exists between two peers, and an accessible relationship between the two mediated peers exists, then a query transformation between the two peers is achievable. Let us assume that there are two mediated peers MP_i and MP_j, and that there is an accessibility relationship between them $MP_j \in R_i(MP_i)$. As such, from equation (32) the global theory T_{GPi} that formally describes MP_i contains all the vocabularies for the ontology of mediated peer MP_j. Also, from equation (31), P_j is an element of the set $AGTS$ that represent all the agents for T_{GPi}. Now, let us consider a query $q_1(x)$ posed to mediated peer MP_i. We will also assume that all the mapping necessary to transform the query to a query $q_2(x)$ exist. From equations (33) and (34), the axioms that represent these mappings are added to T_{GPi}. As such, whenever there is an accessibility relation and all the necessary mappings exist, the transformation of the query is guaranteed. And, by definition, the accessibility relation is transitive. As such, if accessibility relations exist between peers i and j, and j and k, then an accessibility relation exists between peers i and k through j. This means, all possible P2P queries will be calculated □.

3- If a data source S_{ik} belongs to the local network of mediated peer MP_i and a local mapping between the peer and the data source exists, then a query $q_1(x)$ over the mediated peer MP_i is transformable to a local query $q_2(x)$ over the data source S_{ik}: From equations (40) and (43), if the data source S_{ik} belongs to the local network of mediated peer MP_i, then, the local answer at source S_{ik} is part of the global answer at a mediated peer MP_i. Also, since each peer MP_i is formalized by two theories; a global theory T_{GPi} and a local theory T_{LPi}. The local theory represents each mediated peer's local network. From equations (37) and (38), the there are assertions in each local theory for the local mappings between the mediated peer and all the data sources. If a mapping between the mediated peer a data source exist, the assertion will exist. As such the local transformation of the query is achievable. And so, if the data source is part of the peer's local network and a local mapping exists, the query transformation is guaranteed to be calculated □.

5.8 Limitations of the Proposed Framework

As illustrated in the analysis above, the proposed framework addresses several issues that are not addressed by the systems in the literature. This includes, addressing the distributed nature, dynamic nature, and loosely coupled nature of open environment. The proposed model, however, is designed to consume the mapping between various ontologies in order to answer queries. If the mappings between various ontologies are not found, this will form an obstacle for the proposed framework to function appropriately. Another challenge for the proposed system is the accuracy of the mappings returned by the matching algorithm. It is difficult for the ontology matching algorithm to return accurate mappings. And this is why many ontology matching algorithms are semi-automatic. This means, a human expert needs to interact with the ontology matching algorithm. If the ontology matching algorithm yields inaccurate results, one cannot trust the answers to the queries returned by other information systems.

Chapter 6

6 Conclusion and Future Work

This chapter concludes the proposed work and provides some open issues for future research.

6.1 Conclusion

1- In this Thesis, the extensional and extensional reduction models for describing a conceptualization are critically discussed and analyzed. It was shown that, while the extensional description is suitable for describing certain state of the world, the extensional reduction description is appropriate for describing static world in which there is a fixed set of entities. For information systems, multi-agent systems, and in general, any dynamic system in which entities can enter and leave the system, it is shown that there is a need for intensional description of the conceptualization.

2- An intensional model for describing a conceptualization is proposed. The proposed model is based on the theory of PRP for intensional logic. The advantages of the intensional description are discussed. And, both course-grained and fine-grained descriptions for the conceptualization are provided. Ontology and ontological commitment are also formally defined in light of the proposed intensional description.

3- An intensional model for ontology-driven distributed data integration systems in open environment is proposed. The proposed architecture is Mediated-P2P architecture, and the proposed model is based on Intensional Epistemic Logic. While the use of ontology helps bridging the heterogeneity gap between various data sources, the intensionality of the proposed model accounts for the dynamic and loosely-coupled nature of open environment. The formal intensional semantics for queries and query answering are presented. And the model is proven to be both sound and complete.

4- In order to address the heterogeneity nature of open environment, various ontology matching techniques were thoroughly investigated. It is concluded that, the matching technique used need to be structural. S-Match, in particular, was found to address the needs of open environment. This is because it does generate more complex relations, between ontology elements. For example, the S-Match algorithm can find if an element is a generalization or a specialization of another element. This can be useful when ontologies have different granularity levels.

6.2 Future Work

1- Different types of queries need to be studied separately. This is because different types of query can require special treatment. Proper intensional semantics for the different query types need to be studied.

2- Because, different query types can require different treatment and different algorithm to answer. The soundness, completeness, and complexity of the proposed algorithms used for answering different query types need to be closely examined.

3- Cross database query answering need to be investigated. This will require special matching algorithm that will match a concept, or a query, over the mediated ontology to a query that spans a set of data sources instead of a set of queries each of which is targeting one data source. Instead of having the global answer to be the union of all local answers, this will allow the results returned from one data source to filter the results returned from another data source.

4- In defining the query semantics for the intensional based data integration system, the only metric that is considered is the satisfiability of the query. The semantics for query answering, when other metrics are to be considered, i.e. the execution time, need to be investigated. This will allow more flexibility when various agents have different needs.

5- It is obvious that DL is more suitable for representing ontologies as compared to Frame-Based Languages. This is because of their expressiveness and clean semantics. Investigation is still required to settle the question on whether DL can support the IEL semantics.

6- The proposed framework is demonstrated from a view that is mainly theoretical. A solution need to be discussed for the task of the implementation. This will include the establishment of a domain model with all the necessary components for interaction between various entities in open environment. It will likely have other issues and considerations as compared to the theoretical approaches. This may include, for instance, how to handle a situation when a data source decides to leave the environment after its mediator already drafted a query execution plan.

7- A mechanism for ranking all possible answers to a query needs to be developed. This is because the quality of the answer can depend on several factors. These factors may include; the trust between various agents. The quality of an answer at a certain node can also depend on how far this node is from the root node. The further the node is, the more error prone the mapping can be.

Bibliography

Abiteboul, Serge, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset, and Pierre Senellart. 2011. *Web Data Management*. Cambridge University Press.

Adams, Robert Merrihew. 1974. “Theories of Actuality.” *Noûs* 8 (3): 211–31. https://doi.org/10.2307/2214751.

Adjiman, P., P. Chatalic, F. Goasdoue, M. C. Rousset, and L. Simon. 2006. “Distributed Reasoning in a Peer-to-Peer Setting: Application to the Semantic Web.” *Journal of Artificial Intelligence Research* 25 (February): 269–314. https://doi.org/10.1613/jair.1785.

Ali, Islam, and Hamada Ghenniwa. 2012. “Conceptualization-A Novel Intensional-Based Model.” In *International Conference on Knowledge Engineering and Ontology Development*, 2:257–264. SCITEPRESS.

Ali, Islam, and Hamada Ghenniwa. 2014. “Ontology-Driven Mediated Data Integration in Open Environment.” In *KEOD*, 230–239.

Alkhamisi, Abrar Omar, and Mostafa Saleh. 2020. “Ontology Opportunities and Challenges: Discussions from Semantic Data Integration Perspectives.” In *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*, 134–40. https://doi.org/10.1109/CDMA47397.2020.00029.

Allemang, Dean, and James Hendler. 2011. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier.

Anderson, C. Anthony. 1984. “General Intensional Logic.” In *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*, edited by D. Gabbay and F. Guenthner, 355–85. Synthese Library. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-009-6259-0_7.

Androutsellis-Theotokis, Stephanos, and Diomidis Spinellis. 2004. “A Survey of Peer-to-Peer Content Distribution Technologies.” *ACM Computing Surveys* 36 (4): 335–371. https://doi.org/10.1145/1041680.1041681.

Aparasu, Rajender R. 2011. *Research Methods for Pharmaceutical Practice and Policy*. Pharmaceutical Press.

Baader, Franz, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, eds. 2007. *The Description Logic Handbook: Theory, Implementation and Applications*. 2nd ed. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511711787.

Bacon, John. 2008. “Tropes.” February 27, 2008. https://plato.stanford.edu/archives/win2011/entries/tropes/.

Batini, C., M. Lenzerini, and S. B. Navathe. 1986. “A Comparative Analysis of Methodologies for Database Schema Integration.” *ACM Computing Surveys* 18 (4): 323–364. https://doi.org/10.1145/27633.27634.

Bealer, George. 1979. “Theories of Properties, Relations, and Propositions.” *The Journal of Philosophy* 76 (11): 634–48. https://doi.org/10.2307/2025697.

Bealer, George. 1982. *Quality and Concept*. Oxford University Press. https://philarchive.org.

Bealer, George. 1993. “A Solution to Frege’s Puzzle.” *Philosophical Perspectives* 7: 17–60. https://doi.org/10.2307/2214115.

Bealer, George. 1998a. “Intensional Entities.” In *Routledge Encyclopaedia of Philosophy*, 803–7. London: Routledge.

Bealer, George. 1998b. “Propositions.” *Mind* 107 (425): 1–32. https://doi.org/10.1093/mind/107.425.1.

Bertossi, Leopoldo. 2007. “Virtual Data Integration.” AMW. https://www.fing.edu.uy/inco/grupos/csi/AMW07/presentaciones/Tutorial-Bertossi.pdf.

Besana, Paolo, Dave Robertson, and Michael Rovatsos. 2005. “Exploiting Interaction Contexts in P2p Ontology Mapping.” In *In 2nd International Workshop on Peer to Peer Knowledge Management*.

Borst, W. N. 1999. “Construction of Engineering Ontologies for Knowledge Sharing and Reuse.,” 1.

Bouquet, Paolo, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. 2003. “C-OWL: Contextualizing Ontologies.” In *The Semantic Web - ISWC 2003*, edited by Dieter Fensel, Katia Sycara, and John Mylopoulos, 164–79. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-39718-2_11.

Bouquet, Paolo, Luciano Serafini, and Stefano Zanobini. 2003. “Semantic Coordination: A New Approach and an Application.” In *The Semantic Web - ISWC 2003*, edited by Dieter Fensel, Katia Sycara, and John Mylopoulos, 130–45. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-39718-2_9.

Boyles, R. R., A. E. Thessen, A. Waldrop, and M. A. Haendel. 2019. “Ontology-Based Data Integration for Advancing Toxicological Knowledge.” *Current Opinion in*

Toxicology, Systems Toxicology, 16 (August): 67–74. https://doi.org/10.1016/j.cotox.2019.05.005.

Brachman, Ronald J., and Hector J. Levesque. 2004. *Knowledge Representation and Reasoning*. Elsevier. https://doi.org/10.1016/B978-1-55860-932-7.X5083-3.

Calvanese, Diego, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. 2004. "Logical Foundations of Peer-to-Peer Data Integration." In *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 241–251. PODS '04. Paris, France: Association for Computing Machinery. https://doi.org/10.1145/1055558.1055593.

Calvanese, Diego, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. 2018. "Ontology-Based Data Access and Integration." In *Encyclopedia of Database Systems*, edited by Ling Liu and M. Tamer Özsu, 2590–96. New York, NY: Springer. https://doi.org/10.1007/978-1-4614-8265-9_80667.

Canito, Alda, Paulo Maio, and Nuno Silva. 2013. "An Approach for Populating and Enriching Ontology-Based Repositories." In *2013 24th International Workshop on Database and Expert Systems Applications*, 123–27. https://doi.org/10.1109/DEXA.2013.19.

Castanier, Emmanuel, Remi Coletta, Patrick Valduriez, and Christian Frisch. 2013. "WebSmatch: A Tool for Open Data." In *Proceedings of the 2nd International Workshop on Open Data*, 1–2. WOD '13. Paris, France: Association for Computing Machinery. https://doi.org/10.1145/2500410.2500420.

Chawathe, S., H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. 1994. "The TSIMMIS Project: Integration of Heterogenous Information Sources." In . Tokyo, Japan. http://ilpubs.stanford.edu:8090/66/.

Chen, Jingliang, Dmytro Dosyn, Vasyl Lytvyn, and Anatoliy Sachenko. 2017. "Smart Data Integration by Goal Driven Ontology Learning." In *Advances in Big Data*, edited by Plamen Angelov, Yannis Manolopoulos, Lazaros Iliadis, Asim Roy, and Marley Vellasco, 283–92. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-47898-2_29.

Choi, Namyoun, Il-Yeol Song, and Hyoil Han. 2006. "A Survey on Ontology Mapping." *ACM SIGMOD Record* 35 (3): 34–41. https://doi.org/10.1145/1168092.1168097.

Coletta, Remi, Emmanuel Castanier, Patrick Valduriez, Christian Frisch, DuyHoa Ngo, and Zohra Bellahsene. 2012. "Public Data Integration with WebSmatch." In *Proceedings of the First International Workshop on Open Data*, 5–12. WOD '12. Nantes, France: Association for Computing Machinery. https://doi.org/10.1145/2422604.2422606.

Cruz, Isabel F., Flavio Palandri Antonelli, and Cosmin Stroe. 2009. "AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies." *Proceedings of the VLDB Endowment* 2 (2): 1586–1589. https://doi.org/10.14778/1687553.1687598.

Cruz, Isabel F., Cosmin Stroe, Federico Caimi, Alessio Fabiani, Catia Pesquita, Francisco M. Couto, and Matteo Palmonari. 2011. "Using Agreementmaker to Align Ontologies for OAEI 2011." In *Proceedings of the 6th International Conference on Ontology Matching - Volume 814*, 114–121. OM'11. Bonn, Germany: CEUR-WS.org.

Daswani, Neil, Hector Garcia-Molina, and Beverly Yang. 2003. "Open Problems in Data-Sharing Peer-to-Peer Systems." In *Database Theory — ICDT 2003*, edited by Diego Calvanese, Maurizio Lenzerini, and Rajeev Motwani, 1–15. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-36285-1_1.

Davis, Randall, Howard Shrobe, and Peter Szolovits. 1993. "What Is a Knowledge Representation?" *AI Magazine* 14 (1): 17–17. https://doi.org/10.1609/aimag.v14i1.1029.

De Giacomo, Giuseppe, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. 2018. "Using Ontologies for Semantic Data Integration." In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, edited by Sergio Flesca, Sergio Greco, Elio Masciari, and Domenico Saccà, 187–202. Studies in Big Data. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-61893-7_11.

Egami, Shusaku, Xiaodong Lu, Tadashi Koga, and Yasuto Sumiya. 2020. "Ontology-Based Data Integration for Semantic Interoperability in Air Traffic Management." In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, 295–302. https://doi.org/10.1109/ICSC.2020.00059.

Faria, Daniel, Catia Pesquita, Emanuel Santos, Isabel F. Cruz, and Francisco M. Couto. 2013. "Agreement Maker Light Results for OAEI 2013." In *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, 101–108. OM'13. Sydney, Australia: CEUR-WS.org.

Faria, Daniel, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. 2013. "The AgreementMakerLight Ontology Matching System." In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, edited by Robert Meersman, Hervé Panetto, Tharam Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter De Leenheer, and Deijing Dou, 527–41. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-41030-7_38.

Ferreira, Jade, José Maria N. David, Regina Braga, Fernanda Campos, Victor Ströele, and Leonardo de Aguiar. 2019. "Supporting the Collaborative Research through

Semantic Data Integration." In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 319–24. https://doi.org/10.1109/CSCWD.2019.8791911.

Fitting, Melvin. 2004. "First-Order Intensional Logic." *Annals of Pure and Applied Logic*, Provinces of logic determined. Essays in the memory of Alfred Tarski. Parts IV, V and VI, 127 (1): 171–93. https://doi.org/10.1016/j.apal.2003.11.014.

Fitting, Melvin. 2006. "Intensional Logic." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Spring 2020. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2020/entries/logic-intensional/.

Flesca, Sergio, Sergio Greco, Elio Masciari, and Domenico Saccà. 2017. *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer.

"Freenet." n.d. Accessed June 28, 2020. https://freenetproject.org/.

Frege, Gottlob. 1980. "On Sense and Reference." In *Translations from the Philosophical Writings of Gottlob Frege*. Rowman & Littlefield Pub Inc.

Friedman, Marc, Alon Levy, and Todd Millstein. 1999. "Navigational Plans for Data Integration." In *Proceedings of the 1999 International Conference on Intelligent Information Integration - Volume 23*, 72–78. III'99. Stockholm, Sweden: CEUR-WS.org.

Garcia-Molina, Hector, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey Ullman, Vasilis Vassalos, and Jennifer Widom. 1997. "The TSIMMIS Approach to Mediation: Data Models and Languages." *Journal of Intelligent Information Systems* 8 (2): 117–32. https://doi.org/10.1023/A:1008683107812.

Genesereth, Michael R., and Nils J. Nilsson. 2012. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann.

Gill, Zann. 2012. "User-Driven Collaborative Intelligence: Social Networks as Crowdsourcing Ecosystems." In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, 161–170. CHI EA '12. Austin, Texas, USA: Association for Computing Machinery. https://doi.org/10.1145/2212776.2212794.

Giunchiglia, Fausto, Mikalai Yatskevich, and Pavel Shvaiko. 2007. "Semantic Matching: Algorithms and Implementation." In *Journal on Data Semantics IX*, edited by Stefano Spaccapietra, Paolo Atzeni, François Fages, Mohand-Saïd Hacid, Michael Kifer, John Mylopoulos, Barbara Pernici, Pavel Shvaiko, Juan Trujillo, and Ilya Zaihrayeu, 1–38. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-74987-5_1.

Grau, Bernardo Cuenca, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, et al. 2013. "Results of the Ontology Alignment Evaluation Initiative 2013." In , 61. No commercial editor. https://hal.inria.fr/hal-00918494.

Gruber, Thomas R. 1992. "What Is an Ontology?" Knowledge Systems, AI Laboratory. 1992. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

Gruber, Thomas R. 1993. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition* 5 (2): 199–220.

Gruber, Thomas R. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing?" *International Journal of Human-Computer Studies* 43 (5–6): 907–928.

Guarino, Nicola. 1997. "Understanding, Building and Using Ontologies." *International Journal of Human-Computer Studies* 46 (2): 293–310. https://doi.org/10.1006/ijhc.1996.0091.

Guarino, Nicola. 1998. *Formal Ontology in Information Systems: Proceedings of the First International Conference (FOIS'98), June 6-8, Trento, Italy*. Vol. 46. IOS press.

Guarino, Nicola, and Pierdaniele Giaretta. 1995. "Ontologies and Knowledge Bases." *Towards Very Large Knowledge Bases*, 1–2.

Guarino, Nicola, Daniel Oberle, and Steffen Staab. 2009. "What Is an Ontology?" In *Handbook on Ontologies*, 1–17. Springer.

Halevy, A.Y., Z.G. Ives, Jayant Madhavan, P. Mork, D. Suciu, and I. Tatarinov. 2004. "The Piazza Peer Data Management System." *IEEE Transactions on Knowledge and Data Engineering* 16 (7): 787–98. https://doi.org/10.1109/TKDE.2004.1318562.

Halevy, A.Y., Z.G. Ives, D. Suciu, and I. Tatarinov. 2003. "Schema Mediation in Peer Data Management Systems." In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*, 505–16. https://doi.org/10.1109/ICDE.2003.1260817.

Heimbigner, Dennis, and Dennis McLeod. 1985. "A Federated Architecture for Information Management." *ACM Transactions on Information Systems* 3 (3): 253–278. https://doi.org/10.1145/4229.4233.

Heylighen, Francis. 1999. "Collective Intelligence and Its Implementation on the Web: Algorithms to Develop a Collective Mental Map." *Computational & Mathematical Organization Theory* 5 (3): 253–80. https://doi.org/10.1023/A:1009690407292.

Hoehndorf, Robert, Midori A. Harris, Heinrich Herre, Gabriella Rustici, and Georgios V. Gkoutos. 2012. “Semantic Integration of Physiology Phenotypes with an Application to the Cellular Phenotype Ontology.” *Bioinformatics* 28 (13): 1783–89. https://doi.org/10.1093/bioinformatics/bts250.

Huebsch, Ryan, Brent N. Chun, Joseph M. Hellerstein, Boon Thau Loo, Petros Maniatis, Timothy Roscoe, Scott Shenker, Ion Stoica, and Aydan R. Yumerefendi. 2005. “The Architecture of PIER: An Internet-Scale Query Processor.” In *CIDR*.

Hull, Richard, and Roger King. 1987. “Semantic Database Modeling: Survey, Applications, and Research Issues.” *ACM Computing Surveys* 19 (3): 201–260. https://doi.org/10.1145/45072.45073.

Ives, Zachary G., Daniela Florescu, Marc Friedman, Alon Levy, and Daniel S. Weld. 1999. “An Adaptive Query Execution System for Data Integration.” *ACM SIGMOD Record* 28 (2): 299–310. https://doi.org/10.1145/304181.304209.

Jiang, Yue J. 1993. “An Intensional Epistemic Logic.” *Studia Logica: An International Journal for Symbolic Logic* 52 (2): 259–80.

Jubien, Michael. 1972. “The Intensionality of Ontological Commitment.” *Noûs* 6 (4): 378–87. https://doi.org/10.2307/2214312.

Jubien, Michael. 1974. “Ontological Commitment to Particulars.” *Synthese* 28 (3/4): 513–31.

Jubien, Michael. 1975. “Ontological Commitment to Kinds.” *Synthese* 31 (1): 85–106.

Jubien, Michael. 1988. “Problems with Possible Worlds.” In *Philosophical Analysis: A Defense by Example*, edited by David F. Austin, 299–322. Philosophical Studies Series. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-009-2909-8_18.

Kama, Ariane Assélé, Audi Primadhanty, Rémy Choquet, Douglas Teodoro, Frank Enders, Catherine Duclos, and Marie-Christine Jaulent. 2012. “Data Definition Ontology for Clinical Data Integration and Querying.” *MIE*. https://doi.org/10.3233/978-1-61499-101-4-38.

Kambhampati, Subbarao, Eric Lambrecht, Ullas Nambiar, Zaiqing Nie, and Gnanaprakasam Senthil. 2004. “Optimizing Recursive Information Gathering Plans in EMERAC.” *Journal of Intelligent Information Systems* 22 (2): 119–53. https://doi.org/10.1023/B:JIIS.0000012467.66268.9e.

Kementsietsidis, Anastasios, Marcelo Arenas, and Renée J. Miller. 2003. “Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues.” In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, 325–336. SIGMOD ’03. San Diego, California: Association for Computing Machinery. https://doi.org/10.1145/872757.872798.

Lambrecht, Eric, Subbarao Kambhampati, and Senthil Gnanaprakasam. 1999. "Optimizing Recursive Information Gathering Plans." In *IJCAI*, 99:1204–1211.

Lenzerini, Maurizio. 2002. "Data Integration: A Theoretical Perspective." In *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 233–246. PODS '02. Madison, Wisconsin: Association for Computing Machinery. https://doi.org/10.1145/543613.543644.

Levy, A., A. Rajaraman, and J. Ordille. 1996. "Querying Heterogeneous Information Sources Using Source Descriptions." Techreport. Stanford InfoLab. 1996. http://ilpubs.stanford.edu:8090/191/.

Lewis, David. 1986. "On the Plurality of Worlds." *Revue Philosophique de La France Et de l'Etranger* 178 (3): 388–390.

Li, Ren, Tianjin Mo, Jianxi Yang, Shixin Jiang, Tong Li, and Yiming Liu. 2020. "Ontologies-Based Domain Knowledge Modeling and Heterogeneous Sensor Data Integration for Bridge Health Monitoring Systems." *IEEE Transactions on Industrial Informatics*, 1–1. https://doi.org/10.1109/TII.2020.2967561.

"LOCKSS." n.d. Accessed June 28, 2020. https://www.lockss.org/.

Louie, Brenton, Peter Mork, Fernando Martin-Sanchez, Alon Halevy, and Peter Tarczy-Hornoch. 2007. "Data Integration and Genomic Medicine." *Journal of Biomedical Informatics*, Bio*Medical Informatics, 40 (1): 5–16. https://doi.org/10.1016/j.jbi.2006.02.007.

Lumineau, Nicolas, Anne Doucet, and Stéphane Gançarski. 2006. "Thematic Schema Building for Mediation-Based Peer-to-Peer Architecture." *Electronic Notes in Theoretical Computer Science*, Proceedings of the International Workshop on Database Interoperability (InterDB 2005), 150 (2): 21–36. https://doi.org/10.1016/j.entcs.2005.11.032.

Maedche, A., B. Motik, L. Stojanovic, R. Studer, and R. Volz. 2003. "Ontologies for Enterprise Knowledge Management." *IEEE Intelligent Systems* 18 (2): 26–33. https://doi.org/10.1109/MIS.2003.1193654.

Majkić, Zoran. 2009. "Intensional First-Order Logic for P2P Database Systems." Edited by Stefano Spaccapietra. *Journal on Data Semantics XII*, Lecture Notes in Computer Science, , 131–52. https://doi.org/10.1007/978-3-642-00685-2_5.

Makhfi, Pejman. 2007. "Introduction to Knowledge Modeling." 2007. http://www.makhfi.com/KCM_intro.htm.

Malone, Thomas W. 2008. "What Is Collective Intelligence and What Will We Do about It." *Collective Intelligence: Creating a Prosperous World at Peace, Earth Intelligence Network, Oakton, Virginia*, 1–4.

Maurin, Anna-Sofia. 2013. “Tropes,” September. https://plato.stanford.edu/archives/spr2014/entries/tropes/.

Medin, Douglas L., and Lance J. Rips. 2005. “Concepts and Categories: Memory, Meaning, and Metaphysics.” *The Cambridge Handbook of Thinking and Reasoning*. https://www.scholars.northwestern.edu/en/publications/concepts-and-categories-memory-meaning-and-metaphysics.

Mendling, Jan, C. Perez de Laborda, and Uwe Zdun. 2005. “Towards Semantic Integration of XML-Based Business Process Models.”

Milo, Tova, Serge Abiteboul, Bernd Amann, Omar Benjelloun, and Fred Dang Ngoc. 2005. “Exchanging Intensional XML Data.” *ACM Transactions on Database Systems* 30 (1): 1–40. https://doi.org/10.1145/1061318.1061319.

Minsky, Marvin. 1974. “A Framework for Representing Knowledge,” June. https://dspace.mit.edu/handle/1721.1/6089.

Mungall, Christopher J., Michael Bada, Tanya Z. Berardini, Jennifer Deegan, Amelia Ireland, Midori A. Harris, David P. Hill, and Jane Lomax. 2011. “Cross-Product Extensions of the Gene Ontology.” *Journal of Biomedical Informatics*, Ontologies for Clinical and Translational Research, 44 (1): 80–86. https://doi.org/10.1016/j.jbi.2010.02.002.

Ng, Wee Siong, Beng Chin Ooi, and Kian-Lee Tan. 2002. “BestPeer: A Self-Configurable Peer-to-Peer System.” In *Proceedings 18th International Conference on Data Engineering*, 272-. https://doi.org/10.1109/ICDE.2002.994726.

Ng, Wee Siong, Beng Chin Ooi, Kian-Lee Tan, and Aoying Zhou. 2003. “PeerDB: A P2P-Based System for Distributed Data Sharing.” In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*, 633–44. https://doi.org/10.1109/ICDE.2003.1260827.

Ngo, Duy Hoa, and Zohra Bellahsene. 2012. “YAM++ : A Multi-Strategy Based Approach for Ontology Matching Task.” In *Knowledge Engineering and Knowledge Management*, edited by Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d’Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, 421–25. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-33876-2_38.

Ngo, Duy Hoa, and Zohra Bellahsene. 2013. “YAM++ Results for OAEI 2013.” In . https://hal-lirmm.ccsd.cnrs.fr/lirmm-01079130.

Parsons, Terence. 1967. “Extensional Theories of Ontological Commitment.” *The Journal of Philosophy* 64 (14): 446–50. https://doi.org/10.2307/2024427.

Phenotype And Trait Ontology. n.d. Accessed June 27, 2020. http://www.obofoundry.org/ontology/pato.html.

Rahm, Erhard, and Philip A. Bernstein. 2001. "A Survey of Approaches to Automatic Schema Matching." *The VLDB Journal* 10 (4): 334–50. https://doi.org/10.1007/s007780100057.

Rayo, Agustín. 2007. "Ontological Commitment." *Philosophy Compass* 2 (3): 428–44. https://doi.org/10.1111/j.1747-9991.2007.00080.x.

Reinhard, R., T. Meisen, T. Beer, D. Schilberg, and S. Jeschke. 2012. "A Framework Enabling Data Integration for Virtual Production." In *Enabling Manufacturing Competitiveness and Economic Sustainability*, edited by Hoda A. ElMaraghy, 275–80. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-23860-4_45.

Richens, Richard Hook. 1956. "General Program for Mechanical Translation between Any Two Languages via an Algebraic Interlingua." In *Report on Research: Cambridge Language Research Unit, Mechanical Translation*. Vol. 3.

Selman, Bart, and Hector J. Levesque. 1993. "The Complexity of Path-Based Defeasible Inheritance." *Artificial Intelligence* 62 (2): 303–39. https://doi.org/10.1016/0004-3702(93)90081-L.

Sheth, Amit P., and James A. Larson. 1990. "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases." *ACM Computing Surveys* 22 (3): 183–236. https://doi.org/10.1145/96602.96604.

Silva, Nuno, and Joao Rocha. 2003. "MAFRA–an Ontology MApping FRAmework for the Semantic Web." In *Proceedings of the 6th International Conference on Business Information Systems*.

S-Match - Semantic Matching. 2014. http://semanticmatching.eu/s-match.html.

Smith, Barry. 2008. "Ontology (Science)." In *Proceedings of the 2008 Conference on Formal Ontology in Information Systems: Proceedings of the Fifth International Conference (FOIS 2008)*, 21–35. NLD: IOS Press.

Smith, Barry, and Christopher Welty. 2001. "FOIS Introduction: Ontology---towards a New Synthesis." In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, .3–.9. FOIS '01. Ogunquit, Maine, USA: Association for Computing Machinery. https://doi.org/10.1145/505168.505201.

Song, Fuqi, Gregory Zacharewicz, and David Chen. 2013. "An Ontology-Driven Framework towards Building Enterprise Semantic Information Layer." *Advanced Engineering Informatics,* Modeling, Extraction, and Transformation of Semantics in Computer Aided Engineering Systems, 27 (1): 38–50. https://doi.org/10.1016/j.aei.2012.11.003.

Spaccapietra, Stefano, Christine Parent, and Yann Dupont. 1992. “Model Independent Assertions for Integration of Heterogeneous Schemas.” *The VLDB Journal* 1 (1): 81–126. https://doi.org/10.1007/BF01228708.

Staab, Steffen, and Rudi Studer. 2010. *Handbook on Ontologies*. Springer Science & Business Media.

Studer, Rudi, V. Richard Benjamins, and Dieter Fensel. 1998. “Knowledge Engineering: Principles and Methods.” *Data & Knowledge Engineering* 25 (1): 161–97. https://doi.org/10.1016/S0169-023X(97)00056-6.

Sütterlin, Thomas, Simone Huber, Hartmut Dickhaus, and Niels Grabe. 2009. “Modeling Multi-Cellular Behavior in Epidermal Tissue Homeostasis via Finite State Machines in Multi-Agent Systems.” *Bioinformatics* 25 (16): 2057–63. https://doi.org/10.1093/bioinformatics/btp361.

Sütterlin, Thomas, Christoph Kolb, Hartmut Dickhaus, Dirk Jäger, and Niels Grabe. 2013. “Bridging the Scales: Semantic Integration of Quantitative SBML in Graphical Multi-Cellular Models and Simulations with EPISIM and COPASI.” *Bioinformatics* 29 (2): 223–29. https://doi.org/10.1093/bioinformatics/bts659.

Taylor, Nicholas E., and Zachary G. Ives. 2006. “Reconciling While Tolerating Disagreement in Collaborative Data Sharing.” In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, 13–24. SIGMOD ’06. Chicago, IL, USA: Association for Computing Machinery. https://doi.org/10.1145/1142473.1142476.

The D2RQ Platform – Accessing Relational Databases as Virtual RDF Graphs. n.d. Accessed June 27, 2020. http://d2rq.org/.

Trentelman, Kerry. 2009. “Survey of Knowledge Representation and Reasoning Systems.” DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA). https://apps.dtic.mil/sti/citations/ADA508761.

Uschold, Michael. 2015. “Ontology and Database Schema: What’s the Difference?” *Applied Ontology* 10 (3–4): 243–58. https://doi.org/10.3233/AO-150158.

Wang, Jianing. 2012. “A Quality Framework for Data Integration.” In *Data Security and Security Data*, edited by Lachlan M. MacKinnon, 131–34. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-25704-9_16.

Wang, Ying Daisy. 2009. “Ontology-Driven Semantic Transformation for Cooperative Information Systems.” University of Western Ontario.

Woll, R., C. Geissler, and H. Hakya. 2013. “Modular Ontology Design for Semantic Data Integration.” In *Proceedings of the 5th International Conference on Experiments/Process/System Modeling/Simulation/Optimization*.

Xue, Yunjiao. 2010. “Ontological View-Driven Semantic Integration in Open Environments.” https://ir.lib.uwo.ca/etd/16.

Xue, Yunjiao, Hamada H. Ghenniwa, and Weiming Shen. 2010. “A Frame-Based Ontological View Specification Language.” In *The 2010 14th International Conference on Computer Supported Cooperative Work in Design*, 228–33. https://doi.org/10.1109/CSCWD.2010.5471972.

Yang, B., and H. Garcia-Molina. 2002. “Improving Search in Peer-to-Peer Networks.” In *Proceedings 22nd International Conference on Distributed Computing Systems*, 5–14. https://doi.org/10.1109/ICDCS.2002.1022237.

Curriculum Vitae

Name:	Islam Ali
Post-secondary Education and Degrees:	Western University London, Ontario, Canada Ph.D., Software Engineering, 2020.
	Suez Canal University Port Said, Egypt M.Sc., Electrical Engineering, 2007.
	Suez Canal University Port Said, Egypt B.Sc., Hon., 2002.
Honours and Awards:	The Top Student Award and Scholarship for four years in a row 1999-2002
	Syndicate of Engineers Top Graduate Aware 2002
Related Work Experience	Teaching Assistant Western University 2007-2012
	Lecturer Assistant Suez Canal University 2007
	Teaching Assistant Suez Canal University 2002 - 2007

Publications:

Islam Ali and Kenneth McIsaac "**Intensional Model for Data Integration System in Open Environment**" submitted KEOD 2020

Islam Ali and Hamada Ghenniwa "**Ontology-Driven Mediated Data Integration in Open Environment."** In KEOD, pp. 230-239. 2014**.**

Islam Ali and Hamada Ghenniwa "**Conceptualization-A Novel Intensional-based Model."** In KEOD, pp. 257-264. 2012**.**

Islam Ali "**Scalable Video Coding Based on 3D Subband."** M.Sc. Thesis, Suez Canal University, 2007**.**

Islam Ali, Randa Atta, Attia Elsaadawi, and Samir Shaheen "**5/2 Motion Compensated Temporal Analysis Based on Lifting"** Sci. Bull. Fac. Eng. Cairo Univ, vol. 53, no. 2, Apr. 2006.

Islam Ali, Randa Atta, Attia Elsaadawi, and Samir Shaheen "**Motion Compensated Temporal Lifting Analysis with Bidirectional Estimation"** Sci. Bull. Fac. Eng. Ain Shams Univ, vol. 41, no. 1, pp. 623-636, Mar. 2006.