University of Crete
Computer Science Department

Data - Quality Metrics in RDF –based PDMSs

Kitsou Georgia

Master's Thesis

March 2008

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Μετρικές Ποιότητας Δεδομένων σε Δυομότιμα Συστήματα Διαχείρισης Δεδομένων βασισμένα στην RDF

Εργασία που υποβλήθηκε από την

Κίτσου Γ. Γεωργία

ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας :

Κίτσου Γεωργία, Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή :

Βασίλης Χριστοφίδης, Αναπληρωτής Καθηγητής, Επόπτης

Γρηγόρης Αντωνίου, Καθηγητής, Μέλος

Γιάννης Τζίτζικας, Επίκουρος Καθηγητής, Μέλος

Δεκτή :

Πάνος Τραχανιάς, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών
Ηράκλειο, Μάρτιος 2008

Data - Quality Metrics in RDF –based PDMSs

Kitsou Georgia

Master's Thesis

Computer Science Department, University of Crete

Abstract

Scientific or educational communities are striving nowadays for highly autonomous infrastructures enabling to exchange queries and integrate (semi-) structured data hosted by peers. In this context, we essentially need a P2P data management system (PDMS), capable of supporting loosely coupled communities of databases in which each peer base can join and leave the network at free will, while groups of peers can collaborate on the fly to provide advanced data management services on a very large scale (i.e., thousands of peers, massive data). A number of recent PDMSs **[1]** recognize the importance of intensional information (i.e., descriptions about peer contents) for supporting such services. Capturing explicitly the semantics of databases available in a P2P network using a schema enables us to (a) support expressive queries on (semi-) structured data, (b) deploy effective methods for locating remote peers that can answer these queries and (c) build efficient distributed query processing mechanisms.

In our work, we consider that peers advertise their local bases using fragments of community RDF/S schemas (e.g., for e-learning, e-science, etc.). These advertisements (i.e., connected subgraphs of classes and properties) are specified by appropriate RDF/S views and they are employed during query routing to discover the partitioning (either horizontal, vertical or mixed) of data in remote peer bases **[2]**. Two are the main challenges in this setting: (a) due to the high distributed nature of a PDMS we build an effective and efficient lookup service for identifying, in a decentralized fashion, which peer views can fully or partially contribute to the answer of a specific query **[3]** and (b) due to the very large number of peers in a that can actually contribute to the answer of a query, an interleaved query PDMS routing and planning will enable us to obtain as fast as possible the first answers from the most relevant peers while the query is further processed by others **[3]**. As a matter of fact, a sequential execution of the routing and planning

phases for a specific query is not a feasible solution in a PDMS context. This interleaved execution not only favors intra-peer processing, which is less expensive than the inter-peer one, but additionally exhibits the benefit of a parallel execution of the query routing, planning and evaluation in different peers. The above query processing technique relies exclusively on the schema information of the view published by a peer.

However, as the number of peers in a PDMS increases and queries become complex (tree or graph), the number of produced plans that need to be optimized and executed with the interleaved query routing and planning becomes huge. To address this issue we need to prune the search space by considering quality metrics of the data hosted by each peer. As a matter of fact, we should be able to rank the peers contributing to a plan, and thus the plans themselves, according to data quality metrics allowing to discard plans producing poor quality query results (according to a threshold either set by the user or the system). To this end we consider data quality metrics such as *coverage*, *density* and *completeness* of the view instances published by the peers w.r.t. the PDMS schema and its virtual instantiation. In particular, the coverage of a peer database is the ratio of the maximum number of fragment instances one expects to obtain through its view (i.e., when all class instances are related through the properties declared by a peer schema fragment) to the total number of instances published by all peers having the same view. Density of a peer database is the ratio of the actual number of fragment instances exported by a peer (i.e., when materializing its view at a certain point of time) to the maximum number of fragment instances considered previously. Finally, completeness of a peer captures the ratio of the actual cardinality of the fragment instances in a peer to the total cardinality of the schema fragment in all the peers that export it. These quality metrics are expected to significantly prune the search space of query planning while they can be easily maintained in a fully decentralized way.

More precisely, in our work we will address the following issues:

- Provide close formulae for estimating the data quality of peer databases and query plans according to the PDMS schema fragments they involve. The coverage of a peer database w.r.t. a PDMS schema fragment is a global quality metric, since it takes into account the cardinality of all peers publishing instances of this fragment. On the other hand, density is a local quality metric, taking into account only the cardinality of the fragment instances of a specific peer. Completeness of a peer database is a global quality metric and it is actually the product of the coverage and density of the peer. The same is

true for the quality of query plans considering larger schema fragments that are composed of the fragments published by the peers. The most challenging issue in this context is the estimation of the overlap between peer databases by considering different assumptions (e.g. disjointness, independence, containment, or quantified overlap) for the instances of the classes and properties specified by their views.

To the best of our knowledge this is the first work addressing data quality issues in RDF-based PDMSs. Previous work has focused exclusively on models for estimating data quality in the context of relational databases. In **[4]**, a framework for estimating *coverage*, *density* and *completeness* of relational sources and plans in a data integration system has been introduced. Our work extends this framework for RDF-based PDMS. Furthermore, we consider a large number of peers cooperating in a fully decentralized way rather than a centralized data integration system with a small number of sources.. Unlike our work, coverage is the only data quality metric considered in **[5]**. We argue that the existence of the other two metrics is important, since coverage cannot give us information about the percentage of null or missing data that a source contains. Completeness is a much more powerful quality metric than the coverage alone, since it combines the estimation about the data that a source can return (coverage) with the estimation about the actual data contained in the source (density). Moreover, we propose several different overlap cases between peers (e.g. disjointness, independence, containment or quantified overlap of the data between peers), while in this work only the case of independence between sources is considered. In **[6]** various quality criteria (e.g. availability or timeliness of a source) are used to select the best possible sources that contribute to a query plan. However, no close formulae are proposed for the estimation of the data quality characteristics (i.e., they are given and not computed). After the best sources have been found, they participate in the planning process, where the produced plans are ranked with respect to quality characteristics and only the best plans are selected for execution. In addition, the issue of overlap, which significantly affects the plan ranking, is not addressed. Finally, in **[7] [8]** dedicated algorithms are used to learn coverage and overlap statistics of sources in a data integration system. Coverage and overlap of data sources are computed as conditional probabilities and as in **[6]** only the case of data independence between the sources is considered. We believe that mining data quality measures in an infeasible solution for PDMS where peers can joint and leave the system at free will.

Supervisor: Prof. Vassilis Christophides

Στην οικογένειά μου

κ στον επόπτη μου

Ευχαριστίες

Πρώτα από όλα θα ήθελα να ευχαριστήσω το Θεό που με αξίωσε να ολοκληρώσω αυτή την εργασία, παρόλες τις ελλείψεις μου σε γνώσεις και ερευνητική εμπειρία.

Από ανθρώπινης πλευράς, θέλω καταρχήν να εκφράσω τη μεγάλη μου ευγνωμοσύνη προς τον επόπτη μου και Αναπληρωτή Καθηγητή κ. Χριστοφίδη Βασίλη. Δεν έχω λόγια να τον ευχαριστήσω για όλη τη γνώση και την εμπειρία του που προσπάθησε να μου μεταδώσει καθ' όλη τη διάρκεια της συνεργασίας μας. Αν και πολυάσχολος, ήταν πάντα δίπλα μου σε οποιαδήποτε δυσκολία σχετικά με την εκπόνηση της εργασίας για να με συμβουλεύσει και να με ενθαρρύνει. Χωρίς την αμέριστη συμπαράστασή του η εργασία αυτή δε θα είχε ολοκληρωθεί. Τον ευχαριστώ μέσα από την καρδιά μου και εύχομαι κάθε ευτυχία και επιτυχία στη ζωή του.

Επίσης, θα ήθελα να ευχαριστήσω πολύ τον Καθηγητή κ. Γρηγόρη Αντωνίου και τον Επίκουρο Καθηγητή κ. Γιάννη Τζίτζικα, οι οποίοι υπήρξαν μέλη της εξεταστικής επιτροπής της παρούσας εργασίας.

I would like to express my gratitude to Mr. Dan Vodislav for his precious help and ideas for my thesis. His propositions helped a lot to correct most of our statements and formulae and to write the final form of my thesis.

Θα ήθελα επίσης να ευχαριστήσω θερμά το Πανεπιστήμιο Κρήτης και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για τις γνώσεις και τις εμπειρίες που μου προσέφερε κατά τη διάρκεια των μεταπτυχιακών μου σπουδών.

Ένα πολύ μεγάλο ευχαριστώ για την υπομονή τους στο Γιώργο τον Κοκκινίδη και το Λευτέρη το Σιδηρουργό, των οποίων τις εργασίες χρειάστηκα για την εκπόνηση της δικής μου. Οι συμβουλές και οι επεξηγήσεις τους ήταν πολύτιμες στην προσπάθειά μου να κατανοήσω το αντικείμενο της εργασίας.

Επιπλέον, ήθελα να ευχαριστήσω μέσα από την καρδιά μου τους γονείς μου Γιώργο και Φωτεινή και την αδερφή μου Βάσω για την αγάπη και τη στήριξή τους όλο τον καιρό των μεταπτυχιακών μου σπουδών και τη βοήθεια που μου προσέφεραν με πολλούς τρόπους.

Εύχομαι η εργασία αυτή να αποτελέσει μια μικρή ανταμοιβή για όλους όσους με βοήθησαν και πίστεψαν στην ολοκλήρωσή της.

Table of Contents

List of Tables

List of Figures

Chapter 1
INTRODUCTION

Scientific or educational communities are striving nowadays for highly autonomous infrastructures enabling to exchange queries and integrate (semi-) structured data hosted by peers. In this context, we essentially need a P2P data management system (PDMS), capable of supporting loosely coupled communities of databases in which each peer base can join and leave the network at free will, while groups of peers can collaborate on the fly to provide advanced data management services on a very large scale (i.e., thousands of peers, massive data). A number of recent PDMSs (**[1],[11]**,**[12],[13],[14],[15]**) recognize the importance of intensional information (i.e., descriptions about peer contents) for supporting such services. Capturing explicitly the semantics of databases available in a P2P network using schema information enables us to (a) support expressive queries on (semi-) structured data, (b) deploy effective methods for locating remote peers that can answer these queries and (c) build efficient distributed query processing mechanisms.

In our work, we consider that peers advertise their local bases using fragments of community RDF/S schemas **[16]** (e.g., for e-learning, e-science, etc.). These advertisements (i.e., connected subgraphs of classes and properties) are specified by appropriate RDF/S views, called RVL views **[10]**, and they are employed during query routing to discover the partitioning (either horizontal, vertical or mixed) of data in remote peer bases **[2]**. Two are the main challenges in this setting: (a) due to the highly distributed nature of a PDMS we need an effective and efficient lookup service for identifying, in a decentralized fashion, which peer views can fully or partially contribute to the answer of a specific query **[2]** and (b) due to the very large number of peers in a PDMS that can actually contribute to the answer of a query, an interleaved query PDMS routing and planning is required to obtain as fast as possible the first answers from the most relevant peers while the query is further processed by others. As a matter of fact, a sequential execution of the routing and planning phases for a specific query is not a feasible solution in a PDMS context. This interleaved execution not only favors intra-peer processing, which is less expensive than the inter-peer one, but additionally exhibits the benefit of a parallel execution of the query routing,

planning and evaluation in different peers. The above query processing technique relies exclusively on the schema information of the view published by a peer **[2].**

However, as the number of peers in a PDMS increases and queries become complex (e.g. tree or graph-shaped), the number of produced plans that need to be optimized and executed with the interleaved query routing and planning becomes huge. To address this issue we need to prune the search space by considering not only cost, but also quality metrics of the data hosted by each peer. Previous work has considered pruning the space of plans either with respect to a cost model or to some quality metric. However, pruning can be even more efficient if both cost and quality metrics are considered at the same time. Moreover, different users may have very different preferences in terms of the importance of different objectives. For instance, some may want to have a few answers really fast, and others may be willing to wait a little longer for more answers or for better quality answers, etc. The way we select which plans to return to the user must be adaptive to such user preferences and support trade-offs among multiple objectives.

As a matter of fact, in addition to distributed cost-based optimization, we should be able to rank the peers contributing to a plan, and thus the plans themselves, according to data quality metrics(**[4],[5],[6],[7],[8],[20],[21],[22],[23],[28],[29],[30],[34]**) allowing to discard plans producing poor quality query results. A threshold combining both cost and data quality metrics could be set either by the user or the system. In this thesis, we consider data quality metrics such as *coverage*, *density* and *completeness* of the view instances published by the peers with respect to the PDMS schema and its virtual instantiation. In particular, the coverage of a peer database with respect to a specific PDMS fragment is the ratio of the maximum number of fragment instances one expects to obtain through its view (i.e., when all class instances are related through the properties declared by a peer schema fragment) to the total number of instances published by all peers advertising the same view. In other words, the coverage of a peer P with respect to a fragment F is an estimation of the maximum percentage of F instances we could obtain from peer P if all the class instances that form fragment F were related to each other. Density of a peer database with respect to a specific fragment F is the ratio of the number of fragment F instances exported by a peer (i.e., when materializing its view at a certain point in time) to the maximum number of F instances considered previously. In fact, density of a peer P with respect to a fragment F is the probability that the instances of the classes which F comprises are related to each other to form instances of fragment F.

Finally, completeness of a peer P with respect to fragment F captures the ratio of the cardinality of the fragment F instances in P to the total cardinality of the schema fragment F in the PDMS. In other words, the completeness of a peer P with respect to a fragment F expresses the probability that a random instance of F in the PDMS belongs to this peer. In the process of estimating these data quality metrics, the notion of overlap of two or more peers with respect to a fragment is important, i.e. the probability that a random instance of this fragment in the PDMS belongs to these peers.

More precisely, in this thesis we made the following contributions:

- Provide close formulae for estimating the data quality of peer databases and query plans according to the RDFS schema fragments they involve. The coverage of a peer database with respect to an RDFS schema fragment is a *global* quality metric, since it takes into account the cardinality of this fragment in the whole PDMS. On the other hand, density of a peer database is a *local* quality metric, taking into account only the cardinality of the fragment instances in a specific peer. Completeness of a peer database is actually the product of the coverage and density of the peer and is also a *global* quality metric. The same metrics can be used to measure the quality of query plans produced by the PDMS optimizer which are composed from different fragments published by the peers. The most challenging issue in this context is the estimation of the overlap between peer fragments by considering different assumptions (e.g. disjointness, independence, containment, or quantified overlap) for the instances of the classes and properties specified by their views.
- Introduce formulae for cardinality estimation of sets of peers and query plans. These formulae provide cardinality estimations for the two most important operators in our framework, i.e. the union and join operator. Using these formulae we can compute an average value for the cardinalities of all the intermediate results of our query plan, which will eventually help us compute the cost in order to decide on the optimal query plan by considering operators reordering and execution.

To the best of our knowledge this is the first work addressing data quality issues in RDF-based PDMSs. Previous work (**[4][5][6][7][8]**) has focused exclusively on models for estimating

data quality in the context of relational databases. Our work extends this framework for RDF-based PDMSs. More precisely, we generalize the formulae proposed in **[4]** for relational (peer and mediator) tables, to an object-oriented data model such as RDF/S. Furthermore, we consider a large number of peers cooperating in a fully decentralized way rather than a centralized data integration system with a small number of peers.

Unlike our work, coverage is the only data quality metric considered in works as (**[7]**, **[8]**,**[21]**), while other works as (**[5],[23],[29]**) define both coverage and overlap between sources and works like (**[6]**,**[22],[34]**) describe various quality criteria but do not suggest any estimation formulae and they do not address the notion of overlap. We argue that the existence of the other two metrics, density and completeness is important, since coverage cannot give us information about the probability of null or missing data that a source contains. Completeness is a much more powerful quality metric than coverage alone, since it combines the estimation about the data that a source can return (coverage) with the estimation about the actual data contained in the source (density). As a consequence, pruning plans with respect to their completeness rather than coverage scores will be more efficient and will return plans of higher quality to the user. We propose formal definitions and estimation formulae for the data quality metrics we address. Moreover, in contrast to **[21]** in which only a few overlap cases are discussed, we propose several different overlap cases between peers (e.g. disjointness, independence, containment or quantified overlap of the data between peers), while the overlap cases of (**[5], [23], [29]**) are closer to ours.

In our work, we estimate an average value for the cardinality of a query answer, unlike other works like **[24]** where a distribution function is used to estimate the cardinality range of queries. Contrary to previous work as (**[7],[8]**) we believe that mining data quality measures is an infeasible solution for PDMSs where peers can join and leave the system at free will. As a result, in the formulae that we present in the following sections we consider that we know some information about the cardinalities of the fragments the peers export and we only make assumptions about their overlap with respect to a specific fragment, i.e. their common instances of this fragment with respect to the total number of instances of this fragment in the PDMS.

This thesis is organized as follows :

In Chapter 2 our framework is presented, and the use of RDF/S in a SON-based P2P system is described. More precisely, we illustrate the use of RQL and RVL to form queries and advertise peer base contents respectively. An example of peer fragments and their instances in the PDMS is

presented. Also, the notion of query/view subsumption is discussed, as well as the notion of interleaved query routing and planning. In Chapter 3 the notion of overlap between the contents of peers is proposed. Various overlap cases are described and formulae are proposed for each overlap case regarding simple or complex fragments, two or more peers, as well as sets of peers. Moreover, formulae for the cardinality estimation of an arbitrary fragment of the PDMS among an arbitrary number of peers and the total cardinality estimation of a PDMS fragment are presented. In addition, we provide formulae for the cardinality estimation of an arbitrary query plan and the overlap estimation between query plans. In the end of the chapter, related work on overlap is discussed. In Chapter 4, data quality metrics (i.e. coverage, density and completeness) are defined in our framework and estimation formulae are proposed both in the case of two or more peers and in the case of simple or complex fragments. Formulae for these data quality metrics in the case of query plans are presented. In addition, some statements about the value range of these metrics in our framework are made. Finally, related work on data quality metrics is discussed and compared with our work. Finally, Chapter 5 summarizes our contributions and illustrates our intentions for future work.

Chapter 2
A FRAMEWORK FOR RDFS-BASED PDMSs

2.1 RDFS SCHEMA OF A PDMS

In our framework, we consider that every peer provides descriptions about information resources that conform to a number of community schemas (e.g., for e-learning, e-services, etc.). Peers employing the same schema to construct such descriptions in their local base belong essentially to the same Semantic Overlay Network (SON) **[11]**. The notion of SONs appears to be an intuitive way to cluster together peers sharing the same model for a particular domain or application for expressing useful queries and exchange information with others. Of course, a peer may belong to more than one SON, depending on the semantics of its base while it may host only a part of the semi-structured descriptions available in the network. In our context, a PDMS is the union of a number of SONs where queries are answered with data residing at peer bases belonging to the same SON as the query.

A natural candidate for representing descriptive schemas about a community domain or application model (ranging from simple structured vocabularies to complex reference models **[17])** is the Resource Description Framework/Schema Language (RDF/S) **[16]**. The Resource Description Framework (RDF) is a general-purpose language for representing information about resources in the World Wide Web. It is based on a directed graph model and it is particularly intended for representing metadata that can be identified on the Web and moreover resources that cannot be directly retrieved on the Web. The basic idea is that a resource (identified by a URI) can be described through a collection of statements forming a so-called RDF description. A specific resource together with a named property and its value is an RDF statement. The value of a property can be another resource or a literal. A literal is either a simple string or another primitive data type. RDF/S schemas are then used to declare vocabularies, i.e., collections of classes and properties that can be used in resource descriptions for a specific application or domain.

RDF is meant for situations in which information about a resource needs to be processed by applications, rather than being only displayed to people. It provides a common framework for

expressing this information so it can be exchanged between applications without loss of meaning. More precisely, RDF provides (i) a standard representation language for Web metadata; and (ii) a schema definition language (RDF/S) to interpret (meta)data using specific class and property hierarchies (i.e., vocabularies). RDF's vocabulary description language, RDF Schema, is a semantic extension of RDF. It provides mechanisms for describing groups of related resources, the relationships between these resources and determines characteristics of other resources, such as the domains and ranges of properties. RDF/S is chosen for representing the schema of P2P databases, since its modeling primitives are the most appropriate for systems where monolithic RDF/S schemas and resource descriptions cannot be constructed in advance and peers may have only incomplete descriptions about the available resources.

In particular, RDF/S (a) enables a modular design of descriptive schemas based on the mechanism of namespaces; (b) allows easy reuse or refinement of existing schemas through subsumption of both class and property definitions; (c) supports partial descriptions since properties associated with a resource are by default optional and repeated and (d) permits super-imposed descriptions in the sense that a resource may be multiply classified under several classes from one or several schemas. The core primitives of RDF/S schemas are classes and properties. Classes describe general concepts or entities. Properties describe characteristics of classes or relationships between classes. Both classes and properties may be related through subsumption. Every property defined in an RDF/S schema has a domain class (i.e., the class that has this property) and a range class (i.e., the value of this property). A property and its domain and range classes form a schema triple, is denoted by (domain(r), r, range(r)). An RDF/S schema is a set of schema triples forming a directed labelled (multi)graph, called in the sequel RDF/S schema graph.

Formally, an RDF/S schema graph is defined in Definition 1 :

Definition 1 *An RDF/S schema graph is a directed multigraph* $GR = (\{C \cup L\}, R, <^c, <^r)$, *where:*

1. *C is a set of nodes labeled with an RDF/S class name.*

2. *L is a set of nodes labeled with a data type (RDF/S literals).*

3. *R is a set of edges* (c_1, r, c_2) *from a node* c_1 *to a node* c_2 *labelled with a property r, where* $domain(r) = c_1$ *with* $c_1 \in C$ *and* $range(r) = c_2$ *with* $c_2 \in C \cup L$.

4. $<^c$ *is a partial order imposed on nodes in C (RDF/S class subsumption).*

5. $<^r$ *is a partial order imposed on edges in P (RDF/S property subsumption).*

In Figure 1, we can see an example of an RDF/S schema defining the SON of our PDMS, which comprises classes (the circular nodes with class names) C1, C2, C3, C4, C5, C6, C7, C8, C9 and C10 that are connected through properties (the solid edges with property names) r1, r2, r3, r4, r5 and r6. Classes C5 and C6 are subsumed of C1 and C2 respectively. The class subsumption is represented by dashed edges, and in the same way is also represented the subsumption of properties, e.g. property r4 is a subproperty of r1. Finally, classes C7 and C8 are subsumed by C5 and C6 respectively.

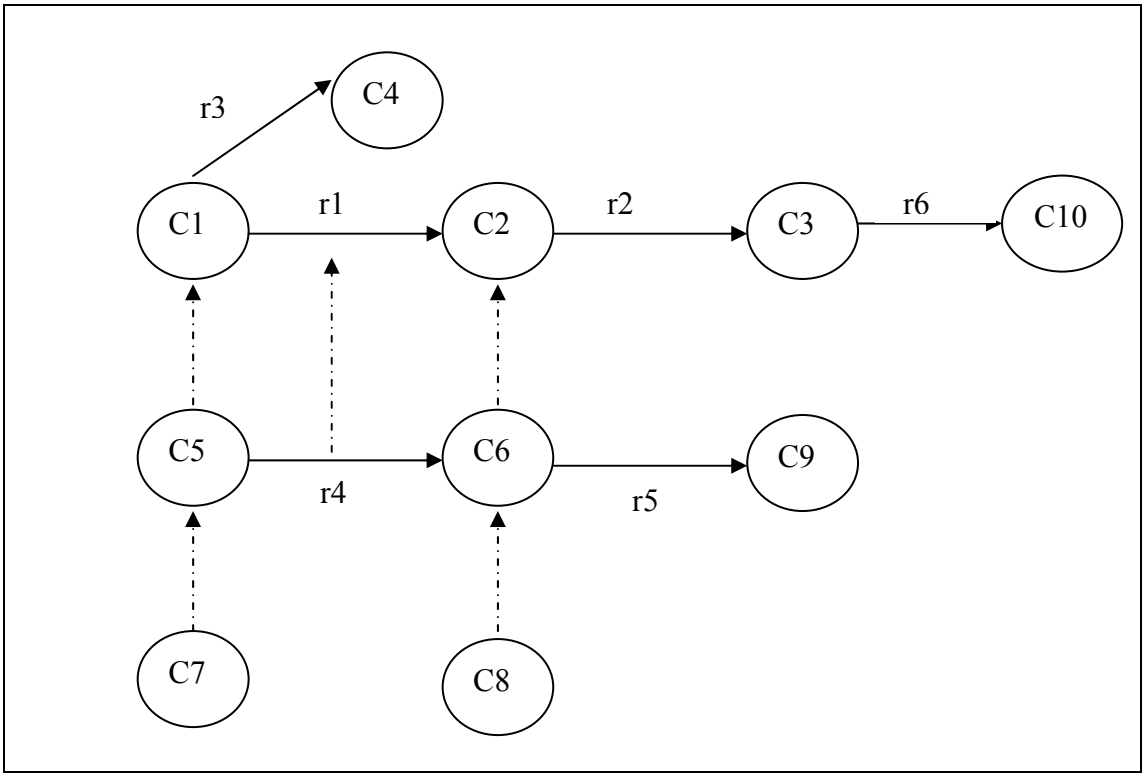


Figure 1 : RDF/S schema of the PDMS

2.2 PEER BASE ADVERTISEMENTS

In the context of a PDMS, each peer should be able to advertise its local base contents to other peers. This is achieved through the views that it exports. Using these advertisements a peer becomes aware of the bases hosted by others in the system. Advertisements may provide descriptive information about the actual data values (extensional) or the actual schema

(intentional) of a peer base. However, since an RDF/S schema may contain numerous classes and properties not necessarily populated in a peer base, we need a fine-grained definition of schema-based advertisements. To this end, we employ views to specify the fragment of an RDF/S schema graph for which all classes and properties are populated in a peer base. In a similar way, peers can retrieve data from the PDMS by issuing queries, which also specify a particular RDF/S schema fragment of interest.

Queries in our framework are formulated in RQL **[18]**, a full-fledged RDF query language which provides sophisticated schema fragment matching facilities against RDF/S schema and data graphs. RQL queries allow us to retrieve the contents of any peer base, namely resources classified under classes or associated to other resources using properties defined in the RDF/S schema. Additionally, peers employ RVL **[10]**, an extension of RQL, for defining views that advertise their contents. RVL views can be either virtual, or materialized. An RVL view specifies the subset of the common schema for which all classes and properties are (in the case of a materialized view) or can be (in the case of a virtual view) populated in a peer base. Both RQL and RVL employ schema fragments to extract the RDF/S schema graph fragments which are relevant to the data requested by a query/view. Table 2.1 summarizes the basic class and property path schema fragments, which can be employed in order to formulate complex RQL/RVL query/view schema fragments (capital letters denote variables, and small letters denote constants). With the exception of the RQL/RVL distinction between exact (denoted with ^) and extended schema fragment matching for class (^c{X} and c{X}) and property ({X}^r{Y} and {X}r{Y}) instances, all the other schema fragments are encountered in the majority of the RDF/S query languages.

In the rest of the thesis we stick on the notion of RDF/S schema fragments, denoted as F, specified by these query/view languages, rather than their syntax on RQL or RVL. A peer view is a subgraph of the PDMS schema graph, called a fragment. A fragment consists of a connected set of classes or properties of the whole schema. A single class can also be regarded as a fragment of the schema, in fact it is the smallest schema fragment. When classes and properties in a fragment are instantiated, the corresponding fragment instances are actually the view data published by the peers of a PDMS. Formally, a schema fragment can be defined as following :

Path Schema fragments	Interpretation
Class Fragments	
$C	{c | c is a schema class}
$C{X}	{[c, x] | c a schema class, x in the interpretation of class c}
$C{X;$D}	{[c, x, d] | c, d are schema classes, d is a subclass of c, x is in the interpretation of class d}
Property Path Schema fragments	
@R	{ r| r is a schema property}
{X} @R {Y}	{[x, r, y] | r is a schema property, [x, y] in the interpretation of property r}
{$C} @R {$D}	{[c, r, d] | r is a schema property, c, d are schema classes, c is a subclass of r's domain, d is a subclass of r's range}
{X; $C} @R {Y; $D}	{[x, c, r, y, d] | r is a schema property, c, d are schema classes, c is a subclass of r's domain, d is a subclass of r's range, x is in the interpretation of c, y is in the interpretation of d, [x, y] is in the interpretation of r}

Table 2.1 : RQL class and property schema fragments

Definition 2 *Given an RDF/S schema graph GR = ({C U L}, R, <c, <r), a fragment specified by a query or view schema fragment over GR is a subgraph GR'= (C', R') such that C'⊆ C and R'⊆ R.*

Table 2.2 illustrates the fragments of the SON RDF/S schema graph specified by the schema fragments of Table 2.1. More precisely, the schema fragment c{X} can be used to retrieve all classes that are instances of class c or any class subsumed by c, while ^c{X} consider only classes that are in the exact interpretation of class c (no subsumed classes are considered). The schema fragment {X}r{Y} can be used to retrieve all the instances (X,Y) of the domain and range classes of property r. Note that this schema fragment takes also into account the class and property subsumption relationships (denoted by the dashed triangles) to include in the result transitive instances of domain/range classes. Schema fragment {X}^r{Y} is similar to the previous, with the exception of considering only the exact interpretation of property r (i.e., no properties subsumed by r will be included in the result).

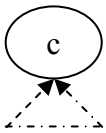
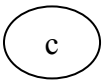
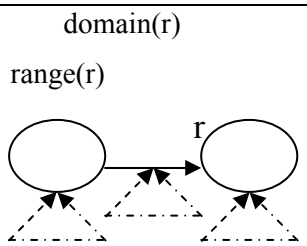
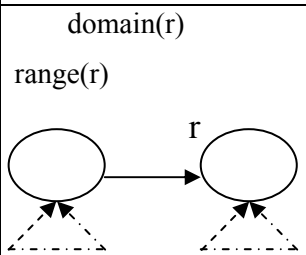
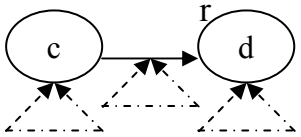
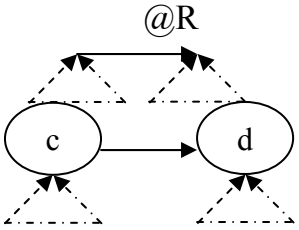
Schema Fragment	Graph Fragment	Schema Fragment	Graph Fragment
c{X}		^c{X}	
{X}r{Y}		{X}^r{Y}	
{X; c}r{Y; d}		{X; c}@R{Y; d}	

Table 2.2 : Graph fragments specified by query/view schema fragments

The next schema fragment, {X;c}r{Y;d}, retrieves all the instances (X,Y) of the class c and d, where c and d are subclasses of the domain and range class of property r respectively. Note that c, r or d can appear with a leading ^ donating the exact interpretation, and if so, the dashed triangle will be omitted in each of the corresponding class or property. Finally, the schema fragment {X;c}@R{Y;d} will return all the properties relating instances of the classes c and d, respectively. These properties can be either defined to have c and d as domain and range classes respectively but also any of the classes subsuming them. Again, the classes can appear in the schema fragment with a leading ^.

We can easily observe the similarity in the intentional representation of both peer base advertisements and query requests as RDF/S schema graph fragments. By representing in the same logical framework what data are requested by a SON (i.e., queries) and what data are actually hosted in each peer base of the SON (i.e., views), we can easily understand the data partitioning (horizontal, vertical, mixed) in remote peers relative to a query.

In Figure 3 we give some examples of the fragments exported by each peer of the PDMS. It is clear that a peer can either export simple fragments (e.g. a single property) or more complex fragments (e.g. properties joined together on a common class).

Apart from the view of the whole PDMS schema fragment, a peer also exports all of its subviews. This means that a peer cannot only answer a query schema fragment that corresponds to its view, but can also contribute to the answer of a query that subsumes its view. In order to decide which peer advertisements match a SON query, we need to check whether the classes and properties of the RDF/S schema fragments specified by the corresponding peer views are subsumed by those of the query. The definition of subsumption between two RDF/S fragments is given below:

Definition 3 *Let the RDF/S schema graph GR = (C, L, R, <c, <r). Let also GR'= {C1, R1) and GR'' = (C2, R2) be two fragments of GR, specified by a query Q and a view V, respectively (C1, C2 $\subseteq$ C and R1, R2 $\subseteq$ R). Q subsumes V (or V is subsumed by Q) if :*

1. $\forall$ *c1* $\in$ *C1,* $\ni$ *c2* $\in$ *C2, c1 = c2 or c2* $<^c$ *c1, and*

2. $\forall$ *r1* $\in$ *R1,* $\ni$ *r2* $\in$ *R2, r1 = r2 or r2* $<^r$ *r1.*

Notice that in the above definition, all classes/properties in Q must be present or subsume a class/property in V. However, V may have additional classes and properties. In the following, both peer views and query fragments will be denoted only as F. Figure 2 illustrates two different subsumption cases.

As shown in Figure 2, peer P2 exports property r4, with domain class C5 and range class C6. Let us assume that the query in our case is the view of r4. Peer P2 exports property r4, which is a subproperty of r1, with domain class C5 and range class C6, which are subclasses of C1 and C2 respectively. So, P2 can answer not only queries that correspond to r4, but also queries that correspond to r1, i.e. r1 horizontally subsumes r4. On the other hand, peer P6 exports properties r4 and r5, which are joined on their common class C6. This view can be employed to answer not only a query concerning this view, but also any of its fragments. In other words, P6 can contribute to the answer of either {X;C5}r4{Y;C6} or {Y;C6}r5{Z;C9}. Peer P6 can also contribute to the query {X;C1}r1{Y;C2}, since r1 horizontally subsumes r4.

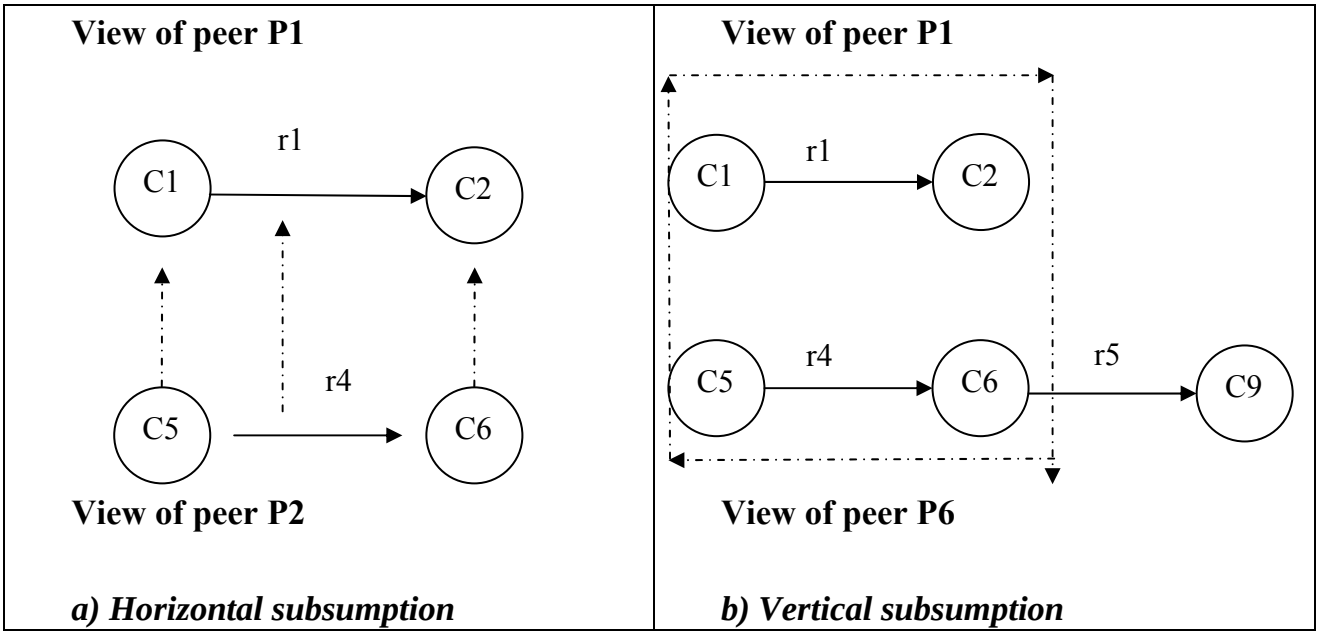
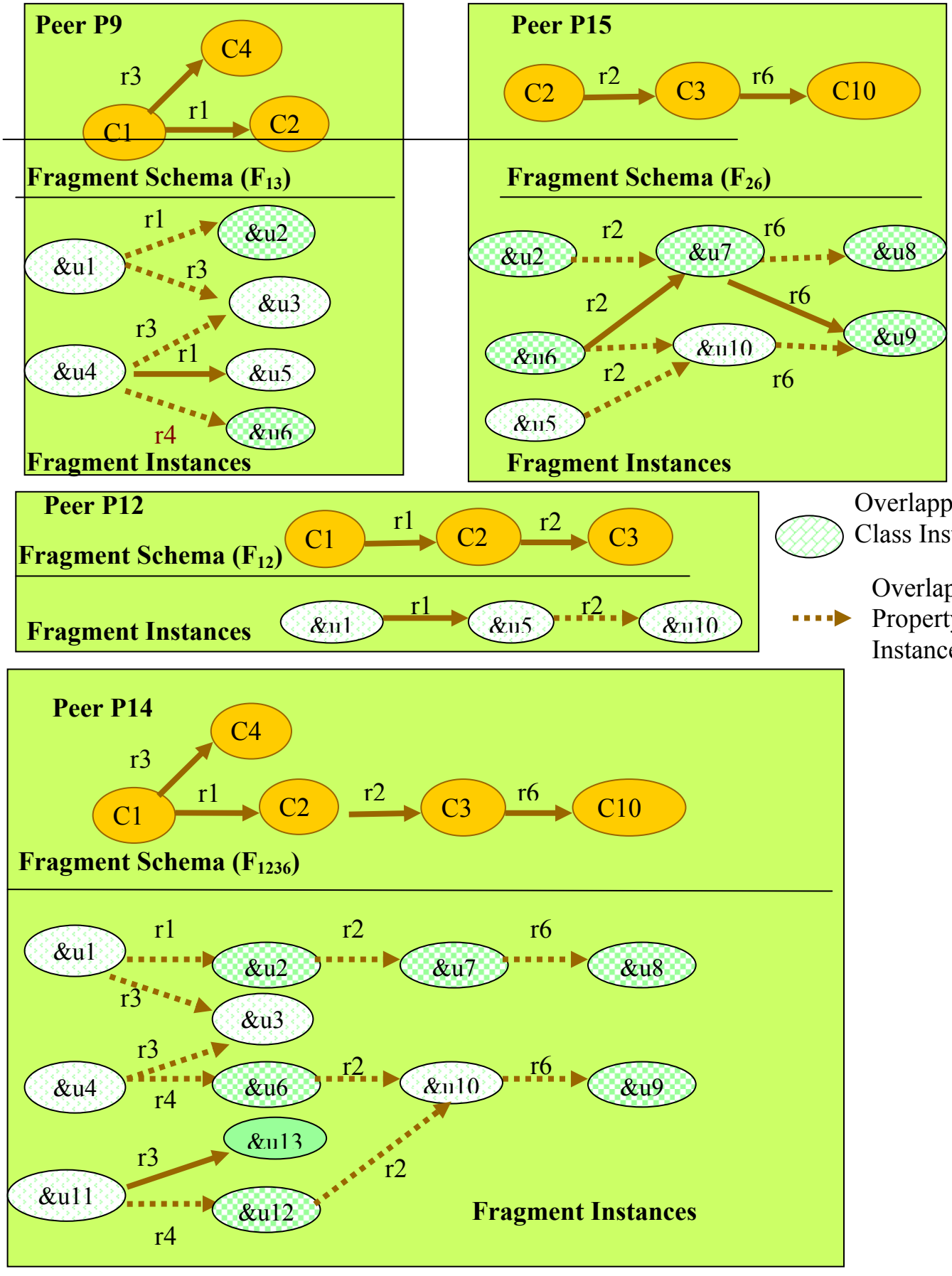


Figure 2 : Vertical/Horizontal query/view subsumption

2.3 PEER FRAGMENTS EXAMPLE

To give an example of the fragments exported by peers of the PDMS and the fragment instances that each peer contains, let us consider the peers P9, P12, P13, P14 and P15 as illustrated in Figure 3. According to the example of Figure 1, class C6 is a subclass of C2 and class C7 is a subclass of C1 while property r4 is a subproperty of r1. Peer P9 exports properties r1 and r3 joined on class C1. We will denote this fragment as F_{13} = r3 ⋈ r1. In the same way, peer P12 exports fragment F_{12} = r1 ⋈ r2, peer P13 exports fragment F_{42} = r4 ⋈ r2, peer P14 exports fragment F_{1236} = r3 ⋈ r1 ⋈ r2 ⋈ r6 and peer P15 exports fragment F_{26} = r2 ⋈ r6. Figure 3 shows the fragment instances that each peer contains.

As we can see, all peers of Figure 3 have common class or property instances. In other words, the bases of peers P9, P12, P13, P14 and P15 overlap with respect to specific schema classes and properties. For example, the instance &u1 of class C1 is common in peers P9, P12 and P14 (depicted by dashed cycles), while peers P12 and P15 have one common instance of property r2 (depicted by dashed arrows), which connects instances &u5 and &u10 of classes C2 and C3 respectively. We will discuss the notion of overlap thoroughly in Chapter 3.


Peer P9
C4
r3
C1
r1
C2
Fragment Schema (F_{13})
r1
&u2
&u1
r3
&u3
r3
r1
&u4
&u5
r4
&u6
Fragment Instances
Peer P15
C2
r2
C3
r6
C10
Fragment Schema (F_{26})
r2
r6
&u2
&u7
&u8
r2
r6
&u6
&u10
&u9
r2
r6
&u5
Fragment Instances
Peer P12
Fragment Schema (F_{12})
C1
r1
C2
r2
C3
Fragment Instances
&u1
r1
&u5
r2
&u10
Overlapped Class Instances
Overlapped Property Instances
Peer P14
C4
r3
C1
r1
C2
r2
C3
r6
C10
Fragment Schema (F_{1236})
&u1
r1
r2
r6
&u2
&u7
&u8
r3
&u3
r3
&u4
r2
r6
&u6
&u10
&u9
r4
&u13
r3
r2
&u11
&u12
r4
Fragment Instances

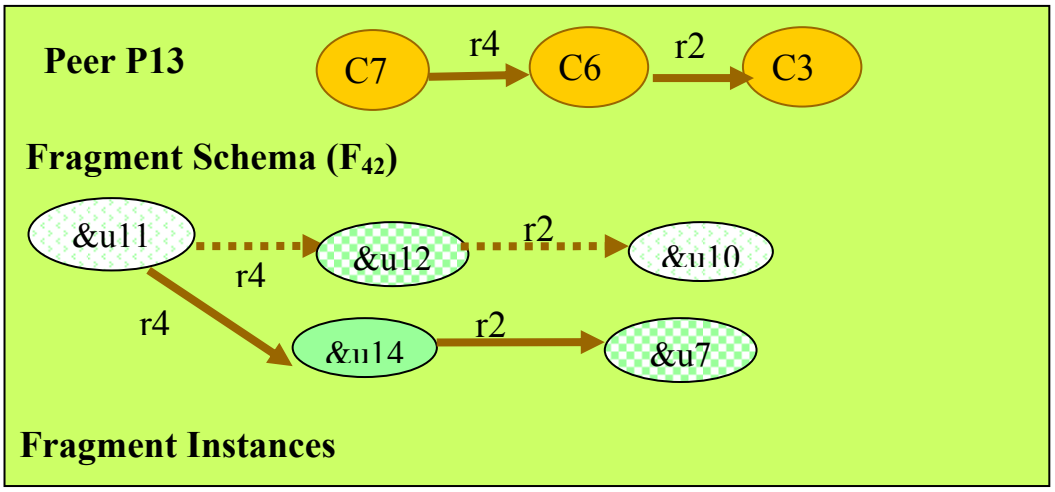


Figure 3 : PDMS schema fragments and their instances in peers

2.4 QUERY ROUTING AND PLANNING

Query processing in our framework is responsible for generating distributed query plans guiding the execution of the query in the PDMS. The query planning phase is based on considering the relevant to the query peer views gathered during the routing phase. The produced query plan specifies precisely how the query is going to be deployed and executed at the selected peers contributing to the final answer by establishing appropriate communication channels **[1]** among them. We distinguish between two scenarios concerning the execution of both query routing and planning algorithms in our context. The first scenario involves the sequential execution of the query routing and planning phases, where the user requires the creation of a single query plan offering complete results to the input query. This way, the produced query plan contains all the simple properties of the query and evaluates the answer by considering the most beneficial ordering of all the joins between them. The second scenario is based on an interleaved execution of the query routing and planning algorithms. It is more complicated, but has several advantages over the sequential scenario. While the first scenario requires a single execution of both query routing and planning phases and a simple fragmentation for the input query, this is not adequate for this second scenario, where more complex fragmentation policies should be considered and many plans should be constructed and executed at several steps.

The interleaved query routing and planning involves a query fragmentation phase where the query is split into distinguished fragments, each of which will eventually be executed as a whole by the same peer. Recall that a query is always a fragment graph of the underlying PDMS schema graph. A component called fragmentor **[1]** is employed in order to split the query based on the

existing data distribution, since the fragments should exploit as much as possible the answering capabilities of the involved peers. There are multiple ways of breaking a view into a specific number of fragments. A set of fragments that when combined produce the complete PDMS schema (or a query in general) is considered as a fragmentation of the schema (or the query). Given a specific number of joins, a set of all the possible fragmentations can be produced. The fragmentor takes as input the number of joins which are required between the produced fragments in order to evaluate the original query. The output is all the possible fragmentations of the query for a specific number of joins.

As we can see in Figure 4, in the simplest case the number of joins equals 0, so the whole view is returned. At each next round of the interleaved routing and planning the number of joins is increased by 1, until no further fragmentation of the view is possible, i.e., the query is decomposed to its primitive components, i.e. simple properties and all the joins involved in it are considered. Let us take as example the schema fragment F_{1236} that peer P14 exports according to our example of Figure 3. This is a fragment of size 4 comprising properties r1, r2, r3 and r6. It is clear that only peer P14 can answer a query requesting F_{1236} as a whole (i.e. fragmentation with 0 joins at round 0). However, by considering all possible fragmentations of F_{1236} (at a round >0) additional anwers to the query can be computed by combining the subfragments published by other peers. Let us assume that at a specific point in time the only peers of the PDMS. are those depicted in Figure 3. and let us call Fij…k the sub-fragment composed of properties ri, rj, … rk. When the number of joins is 1 (round 1), all the possible fragmentations we can obtain are $F_{123} \bowtie F_6$, $F_{126} \bowtie F_3$ and $F_{13} \bowtie F_{26}$. When it comes to a fragmentation with 2 joins (round 2), all the possible fragmentations are $F_{12} \bowtie F_3 \bowtie F_6$, $F_{26} \bowtie F_1 \bowtie F_3$ and $F_{13} \bowtie F_2 \bowtie F_6$. Finally, in the case of a fragmentation with 3 joins (round 3), only fragmentation $F_1 \bowtie F_2 \bowtie F_3 \bowtie F_6$ is possible. In order to discover which peer can be ansewer a specific sub-fragment, query routing is taking palce.

Query routing takes into account the data distribution (horizontal, vertical or mixed) of peer bases committing to the PDMS schema. Query routing is responsible for the data localization **[1]** and the lookup phases **[2]**. The lookup service is responsible to identify and return the peer advertisements relevant to each fragment of the query, with the possible help of a peer view index embedded in the PDMS. Then, the data localization phase matches a given RQL query against a set of RVL peer views in order to determine relevant peer bases based on appropriate query/view

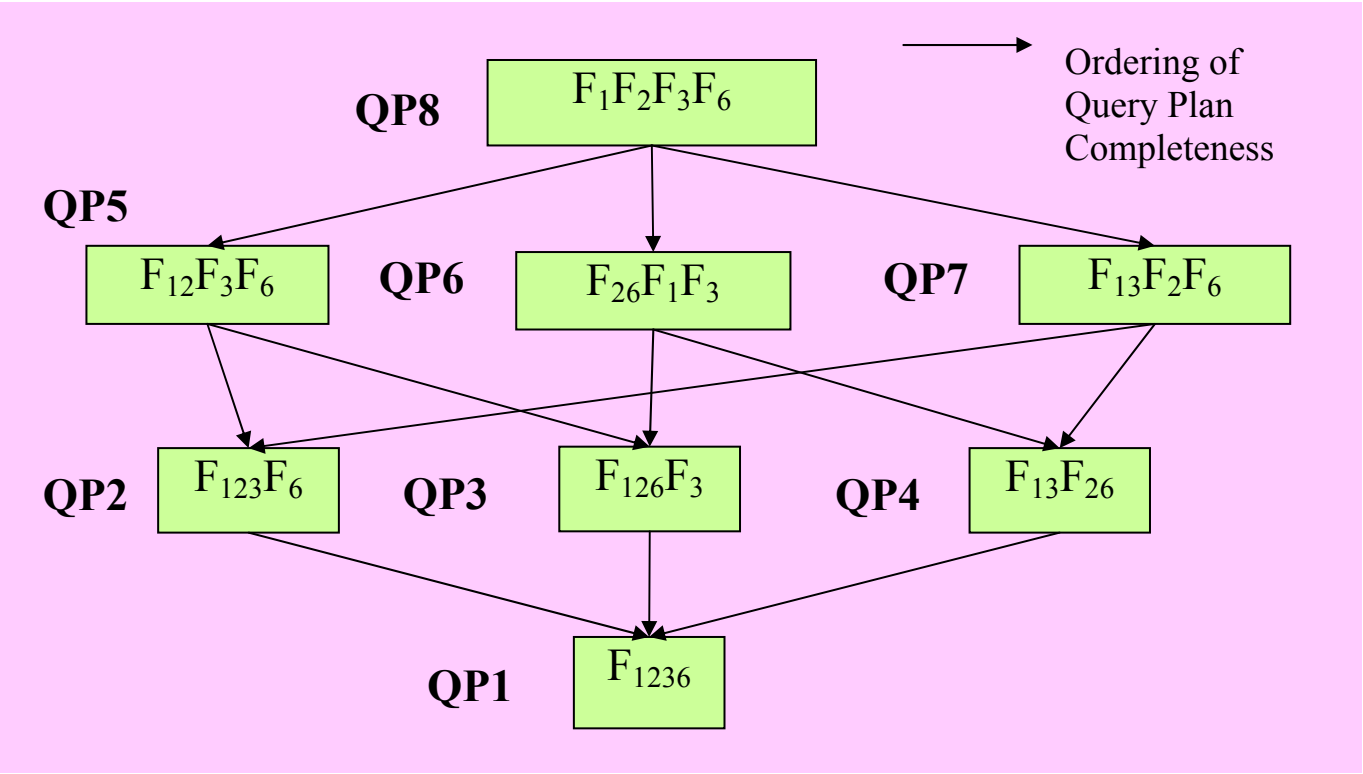


Figure 4: Possible fragmentations of a query schema fragment

subsumption techniques. In fact, the data localizatio n algorithm annotates each schema fragment of the query with the peers that can answer it. As already stated, when a peer P exports a PDMS fragment, it also exports all of its sub – fragments. The peers that can answer each sub – fragment of F, as well as the localization information of the sub - fragments are shown in Table 2.3.

Fragment	Localization information
F_{1236}	P14
F_{123}	P14
F_{126}	P14
F_{13}	P14, P9
F_{26}	P14, P15
F_{12}	P14, P13, P12
F_1	P9, P12, P13, P14
F_2	P12, P13, P14, P15
F_3	P9, P14
F_6	P14, P15

Table 2.3 : Query fragments annotated with localization information

Query planning in our framework is responsible for generating a distributed query plan according to the localization information returned by the routing algorithm. There are two operators used in a PDMS: the join operator and the union operator. If more than one peer can answer the same query, the results from each such peer base are "unioned" (horizontal distribution). As the query is traversed, the results obtained for different queries that are connected at a specific domain or range class are "joined" (vertical distribution) **[3]**. The final query plan is created when all its consistent fragments are translated.

Since for almost every fragment of Figure 4 there are two or more peers that export it, we can write each query plan in two equivalent ways: either as a union of joins of the sub – fragments that form the specific query plan, or as joins of the unions of the peers that export each sub – fragment. Let us consider the join between the sub – fragments F_{123} and F_6. In the first case, the query plan is written as $F_{123} \bowtie F_6$ which means $(F_{123} \bowtie F_6)\Sigma_{Pi}$, i.e. it is evaluated on the set of all the peers in the PDMS. This "union of joins" of fragment instances found on individual peers, i.e. $\sum ij((F_{123})_{Pi} \bowtie (F_6)_{Pj})$, where Σ means here union, may be written as a "join of unions" plan. Thus, in the second case, the query plan can be written as $(F_{123})\Sigma_{Pi} \bowtie (F_6)\Sigma_{Pj}$, i.e. the join between the set of F_{123} instances found in the whole PDMS with the set of F_6 instances in the whole PDMS. Notice that results produced by plan F_{1236} are included in those produced by, e.g. $F_{123} \bowtie F_6$, because $F_{1236} = \sum i((F_{1236})Pi) = \sum i((F_{123})Pi \bowtie (F_6)Pi) \subseteq \sum ij((F_{123})Pi \bowtie (F_6)Pj) = F_{123} \bowtie F_6$. More generally, replacing a fragment by a join of sub-fragments adds new results to those produced by the original plan. This inclusion relation induces a partial order relation among the above plans in terms of query answers, as illustrated in Figure 4. Arrows in the lattice connect elements to "smaller" ones; the largest plan is at the top and the smallest one is at the bottom.

As we will discuss in the following, during query routing and planning, the corresponding data quality metrics (i.e. coverage, density and completeness) of each peer that can answer a specific query fragment have been computed. This way, during query planning the peer responsible to create a plan will not need to contact every time the peers that contain data quality information about the subfragments of the plan. The annotated schema fragment of the query produced by the data localization algorithm is then translated to an appropriate algebraic plan using the algebra presented in **[1]**. Figure 5 depicts the plans involving only one join between sub – fragments (i.e. round 1) according to the fragments illustrated in Figure 4 and the peers of Table 2.3.

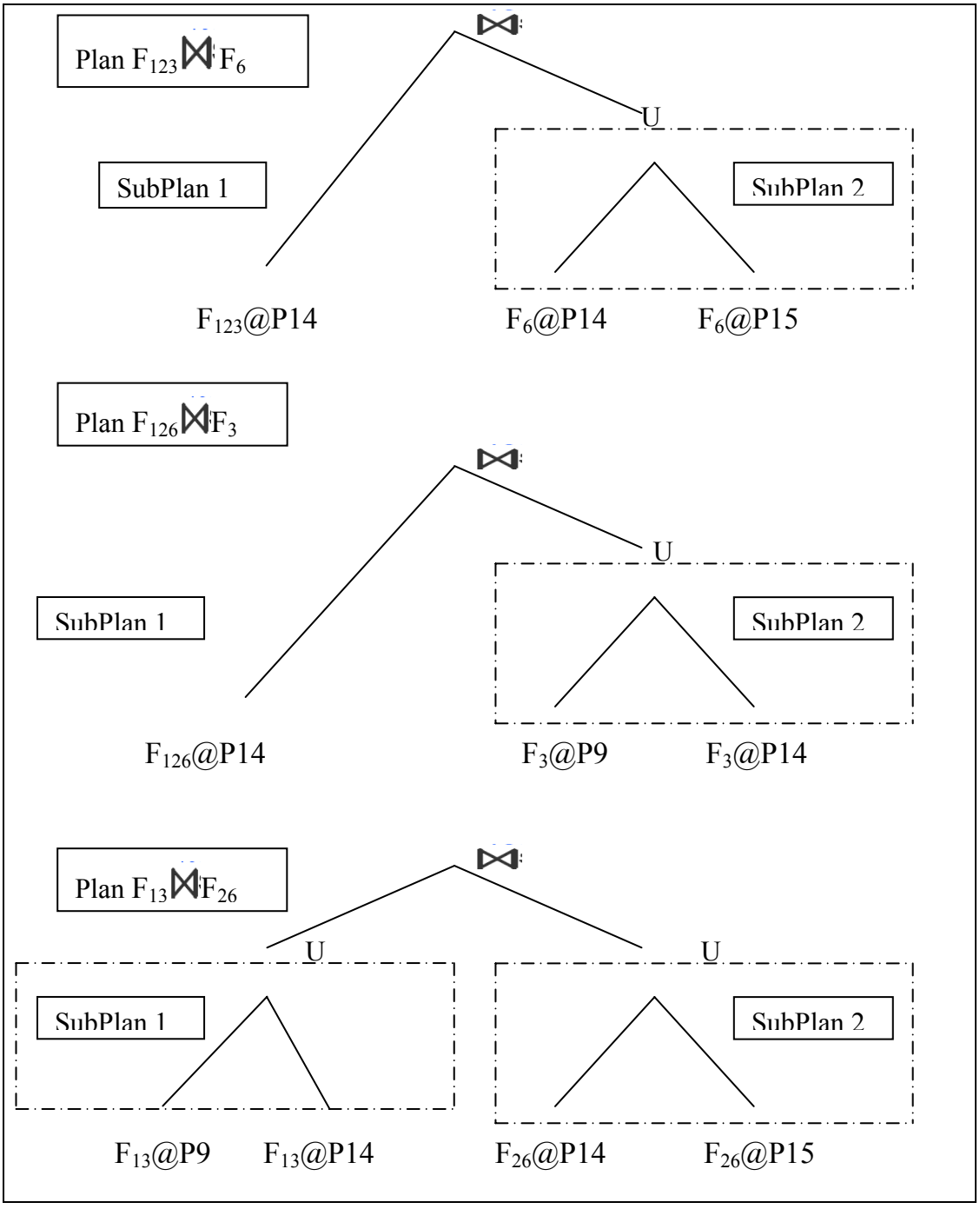


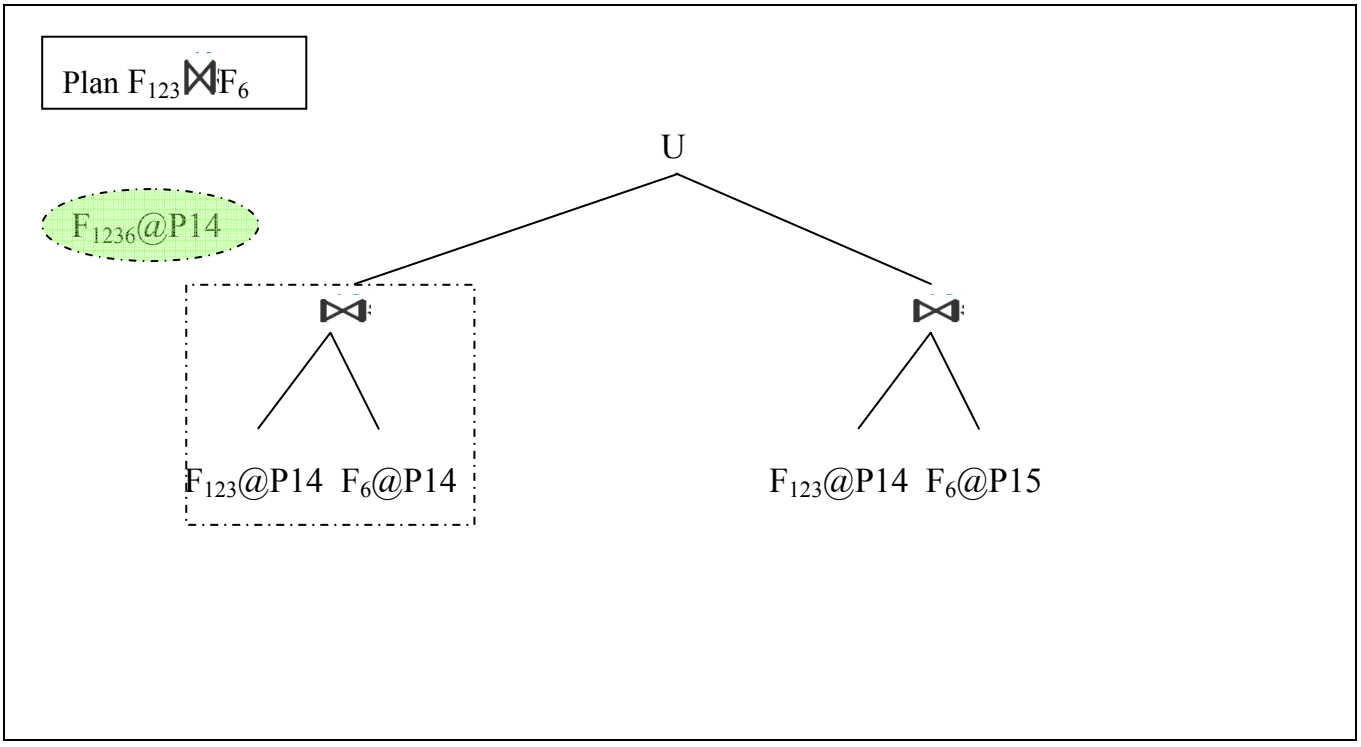
Figure 5 : Query plan creation

Each one of these query plans is passed to an optimizer in order to apply heuristic and/or cost-based optimization techniques **[1]** as long as the number of inter-peer joins in the equivalent query plan is less than the intra-peer one. In the interleaved query routing and planning phase, the only way to achieve completeness and correctness in the results is to push unions at the top of the query plan. This makes possible (a) to evaluate an entire join at a single peer (intra-peer processing) when its view is subsumed by the query fragment, and (b) to parallelize the execution of the union in several peers. As a result, at each consequent phase a number of plans consisting

of only inter-peer (i.e. between different peers) joins exist under each union. This number equals to all the possible plans created from different fragmentations but with the same number of inter-peer joins.

However, at each such fragmentation, if the unions, which combine the data for each fragment from all the peers that can answer it are pushed up, the number of these plans is increased. At each step of the interleaved execution, query plans that are considered at the previous phases of the execution should be removed, since their results are already computed. This actually means that only inter-peer processing is considered, since all emerging intra-peer joins have been handled in the previous phases **[1]**. In this sense, the query plans of Figure 5 can be rewritten as shown in Figure 6.

Finally, the optimized plans are sent to the execution engine responsible for forwarding the already distinguished subplans to the appropriate peers and monitoring their evaluation. Peer communication is achieved by the use of appropriate communication channels that additionally provide the means for query plan adaptation during query execution in case of run-time failures. We can easily observe from our example that taking into account the vertical distribution ensures correctness of query results (i.e., produce a valid answer), while considering horizontal distribution in query plans favors completeness of query results (i.e., produce more and more valid answers).



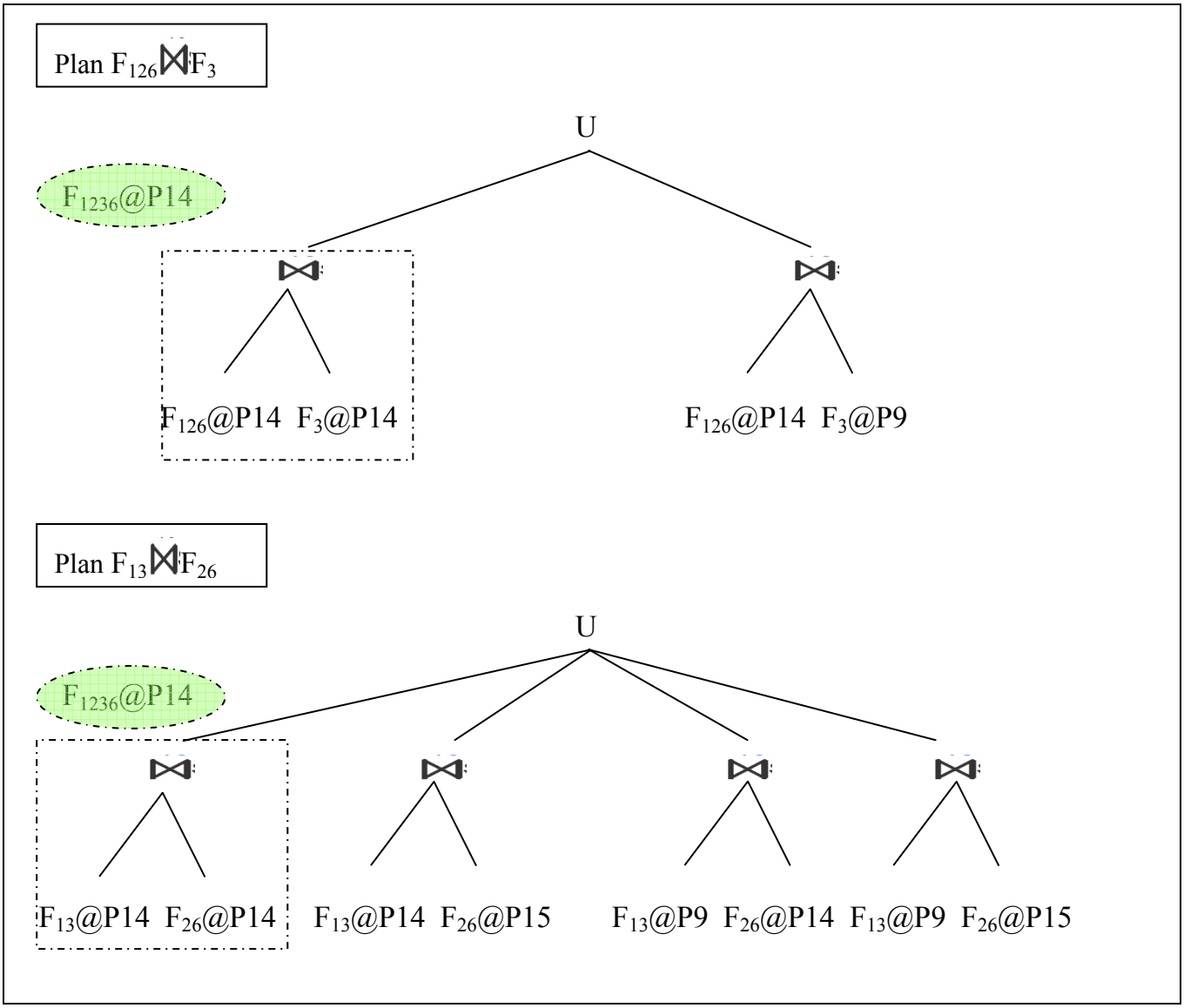


Figure 6 : Optimizing query plans using heuristics and algebraic equivalences

The number of query plans that are generated by the query planning algorithm by considering all fragmentation alternatives of a given query can be fairly large. Pruning the available plans with the use of appropriate quality metrics in addition to estimations about their execution cost is a great challenge. By using data quality metrics such as coverage, density and completeness of the view instances published by the peers with respect to the PDMS schema and its virtual instantiation, we intend to prune as much as possible the planning space and provide users with answers of higher quality, in a reasonable processing time. A threshold combining both cost and data quality metrics could be set either by the user or the system. In the following, we give the definitions of the quality metrics and we propose formulae for their calculation in different cases.

Chapter 3

OVERLAP OF PEER BASES

With the term "overlap" in our context we refer to the degree to which two or more peers contain the same data of a specific fragment (either a single class or property, or a more complex fragment), i.e. export the same instances of a specific fragment. E.g. if two or more peers export the same class, we can estimate their overlap with respect to this class, i.e. their common instances of this class. In the same way we can estimate the overlap of a single property, or even the overlap of a more complex fragment (consisting of two or more properties joined together) among the peers that export it.

We can either use probabilities to express overlap, or it can be expressed as the cardinality of the common instances of two or more peers with respect to a fragment. In the following we will show overlap estimations in both forms. To distinguish between them, we will denote the overlap of two peers Pi, Pj with respect to a fragment F as $overlap_{Pi,Pj}(F)$, when it is expressed as a probability, and as $Overlap_{Pi,Pj}(F)$, when it is expressed as cardinality. The different overlap cases are given below:

- **Independence**: peers Pi and Pj are independent with respect to F if there is no (known) dependency of F instances in peers Pi and Pj. That is, there is some coincidental overlap. As a result, $overlap_{Pi,Pj}(F)$ is estimated. Whenever there is no concrete knowledge about overlap (see the following cases), we assume independence.
- **Disjointness**: it is known that peers Pi and Pj do not have any common instances of fragment F, so, $overlap_{Pi,Pj}(F) = 0$.
- **Quantified overlap**: In some occasions the exact degree of overlap with respect to F is known, i.e., it is known that peers Pi and Pj have X% common F instances, where X is a known probability. Then, $overlap_{Pi,Pj}(F)= X/100$.
- **Containment**: in this case, it is known that the F instances of one of the peers are contained in the F instances of the other or vice versa. Let us assume that the F instances of Pi are included in the F instances of Pj. Thus, $overlap_{Pi,Pj}(F) = 1$.

We should point out that the independence assumption for overlap estimation is more realistic than the other assumptions in an open dynamic setting such as a PDMS. In fact, in the independence case, overlap is the *probability that a random instance of a fragment F belongs to two or more peers*. To compute this probability we should consider a universe of discourse, which is determined by the morphology of fragment F (e.g. a single property, a join of two properties, etc). In our case, we consider the maximum number of F instances expected to be retrieved in our PDMS. This reference set is modeled as the product of the instances of all the classes involved in a fragment and it represents the maximum number of fragment instances we can obtain from our PDMS. As we will observe during overlap estimation the connectivity of resource nodes through property edges increases as peers publish more fragments sharing common resources.

To give an example of overlap, let us consider the peers and their instances in Figure 3. It is obvious that the property instances that connect the same domain and range class instances are in fact the same. In favor of simplicity we will assume that the peers and fragment instances addressed above are the only peers and fragments of the PDMS.

Let us denote the cardinality of each peer P with respect to the fragment F it exports as $|F|_P$ and the total cardinality of fragment F in the PDMS as ||F||, which is the reference set for overlap computations. The cardinality estimation of ||F||, where F is an arbitrary fragment in a PDMS with an arbitrary number of peers, will be presented in sections 3.7 and 3.9. The cardinalities of the peers with respect to the fragments they export, as well as the total cardinalities of the PDMS fragments are shown in Table 3.1. We should note that when computing the total cardinality of a fragment F that involves two or more properties joined together, we should take into account both the joins that concern properties in the same peer and the joins that concern properties that come from different peers. For example, let us consider fragment r2 ⋈ r6. This fragment is exported as a whole by peers P14 and P15. However, instances of fragment r2 ⋈ r6 can occur also by joining instances of r2 and r6 that belong to other peers that export one of these properties. Peers P12, P13, P14 and P15 export property r2, and peers P14 and P15 export property r6. So, the total number of r2 ⋈ r6 instances includes also those instances created by joining fragments of different peers.

Total PDMS fragment cardinalities		
Fragment C1 $\|\|C1\|\| = 3$	**Fragment C2** $\|\|C2\|\| = 5$	**Fragment C3** $\|\|C3\|\| = 2$
Fragment C4 $\|\|C4\|\| = 2$	**Fragment C7** $\|\|C7\|\| = 1$	**Fragment C6** $\|\|C6\|\| = 2$
Fragment C10 $\|\|C10\|\| = 2$	**Fragment r1** $\|\|r1\|\| = 6$	**Fragment r2** $\|\|r2\|\| = 6$
Fragment r3 $\|\|r3\|\| = 3$	**Fragment r4** $\|\|r4\|\| = 2$	**Fragment r6** $\|\|r6\|\| = 2$
Fragment r1 ⋈ r2 $\|\|r1 \bowtie r2\|\| = 8$	**Fragment r3 ⋈ r1** $\|\|r3 \bowtie r1\|\| = 7$	**Fragment r2 ⋈ r6** $\|\| r2 \bowtie r6\|\| = 9$
Fragment r3 ⋈ r1 ⋈ r2 $\|\| r3 \bowtie r1 \bowtie r2\|\| = 8$	**Fragment r1 ⋈ r2 ⋈ r6** $\|\| r1 \bowtie r2 \bowtie r6\|\| = 10$	**Fragment r1 ⋈ r2 ⋈ r3 ⋈ r6** $\|\| r1 \bowtie r2 \bowtie r3 \bowtie r6\|\| = 13$

Table 3.1 : Total fragment cardinalities of the PDMS

Before we give an example of overlap estimation, we should note that we will use the term "overlap" to refer to the overlap expressed as probability and the term "Overlap" to refer to the overlap expressed as cardinality.

As we can see in Figure 3, peers P9, P12 and P14 have a common instance of class C1. The total number of C1 instances in the PDMS is 3. So,

$Overlap_{P9,P12,P14}(C1) = 1$

$overlap_{P9,P12,P14}(C1) = 1 / 3 = 0,33$

Besides, peer P12 has only one C1 instance which is involved in the C1 instances of peer P9. As a result, all the C1 instances of P12 are contained in the C1 instances of P9. Thus :

$overlap_{P12,P9}(C1) = 1$

$overlap_{P9,P12}(C1) = 1 / 2 = 0,5$

Peer P13 exports instances of class C7, which is a subclass of C1, i.e. C1 horizontally subsumes C7. Peers P13 and P14 have one common C7 instance, which in P14 is exported as a C1 instance, since P14 exports class C1. The total number of C1 instances in the PDMS is 3. As a result,

$Overlap_{P13,P14}(C1 \cup C7) = 1$

$overlap_{P13,P14}(C1 \cup C7) = 1 / 3 = 0,33$

Peers P9 and P14 export property r1 with the same domain and range classes, C1 and C2 respectively. They have two common r1 instances, while the total number of r1 instances in the PDMS is 6, so

$Overlap_{P9,P14}(r1) = 2$

$overlap_{P9,P14}(r1) = 2 / 6 = 0,33$

Peer P13 exports instances of property r4, which is a subproperty of r1, i.e. r1 horizontally subsumes r4. The domain and range classes of r4 in P13 are subclasses of the domain and range classes of r1 in P14. Peers P13 and P14 have one common r4 instance, which in P14 is exported as an r1 instance, since P14 exports property r1. The total number of r1 instances in the PDMS is 6. As a result,

$Overlap_{P13,P14}(r1 \cup r4) = 1$

$overlap_{P13,P14}(r1 \cup r4) = 1 / 6 = 0,17$

Peers P14 and P15 both export fragment $r2 \bowtie r6$ with the same domain and range classes. They have two common instances of this fragment, while the total number of $r2 \bowtie r6$ instances in the PDMS is 9. So,

$Overlap_{P14,P15}(r2 \bowtie r6) = 2$

$overlap_{P14,P15}(r2 \bowtie r6) = 2 / 9 = 0,22$

Before we show examples and formulae for overlap computation in each of the above cases, we will present the main assumptions and information that is necessary. As we observed in the example above, an important factor of the overlap computation is the distinct number of the instances of F in the PDMS, i.e. $||F||$. We should note that in the example above we can easily compute the distinct number of instances for each fragment of the PDMS. However, in reality it

is very difficult, as peers can join and leave the system at free will and no peer can hold precise fragment cardinalities. In addition, the existence of class and property subsumption makes it even harder to calculate the total cardinality of a specific fragment in the PDMS. As a result, the existence of duplicates affects our estimations for fragment cardinalities and thus the precision but not the formulae of our metrics. In other words, the factor ||F|| denotes the distinct total number of F instances in the PDMS, but in reality its precise value may be affected by duplicate instances. We will denote as $\sum_i |F|_{Pi}$ the total cardinality of fragment F in the PDMS, considering also duplicates. Then, it holds that ||F|| $\leq \sum_i |F|_{Pi}$. By computing the overlap between peers, we can make corrections to the estimation we have for ||F||. The problem of estimating the total cardinality of an arbitrary fragment of the PDMS schema will be presented thoroughly in sections 3.7 and 3.9. At this point we should note that in the rest of this work for readability reasons we will use the symbol ||F|| instead of the symbol ||F|| to denote the total cardinality of fragment F in the PDMS.

Our main assumption about the PDMS schema classes/properties is that when a peer exports instances of a specific class/property, it exports not only direct, but also transitive instances, i.e. instances of its subclasses/subproperties (horizontal subsumption). For example, peer P1 of Figure 2 exports instances of class C1, and peer P2 exports class C5, a subclass of C1, which has no subclasses. Then, apart from the direct C1 instances that are exported, there may be instances of C5, which are exported as instances of C1, i.e. transitive C1 instances. In fact, all the instances of class C5 can be exported as C1 instances. We will represent the sum of all direct C1 instances in the PDMS as $||\hat{C1}||$. The same assumption holds for properties. Among the instances of a property that a peer exports may well be involved instances of its subproperties.

Let us now generalize our assumptions to the whole PDMS. We denote as ||C1|| and ||C5|| the sum of the C1 and C5 instances (both direct and transitive) respectively that are exported by all the peers of the PDMS. We previously stated that the peers which export C1 instances, also export C5 instances. As a result, transitive C1 instances consist of all C5 instances and thus, under both an open-world and a closed-world assumption of the schema level the following relations hold :

$||C1|| = ||\hat{C1}|| + ||C5|| \Rightarrow ||C1|| > ||C5||$

The same assumption holds for the properties and their subproperties.

Another important element for overlap computation is that each peer of the PDMS knows how many instances of a fragment it exports, whether it is a single class or property, or a more complex fragment. However, when there are two or more peers which export the same fragment there is no way to find the exact number of their common fragment instances. As a result, we use probabilities to estimate overlap.

In the following sections, we will provide examples for the overlap computation of the cases presented above and we will present the formulae we use.

3.1 PEERS PUBLISHING THE SAME CLASS

We will first take the simplest fragment, i.e. a class. Let us consider the simplest case of two peers that export the same class and the probabilities that a random instance of this class appears in all of them are independent with each other (independence overlap case). For example, in Figure 3 peers P9 and P12 export direct instances of class C1. We denote as ||C1|| the sum of all C1 instances in the PDMS. This is the reference set for the overlap estimations with respect to class C1. We also denote as $|C1|_{P9}$ and $|C1|_{P12}$ the cardinalities of C1 in peers P9 and P12 respectively. We want to estimate the probability that a random C1 instance of the reference set is common in both peers, i.e. is a C1 instance of P9 and P12. This is presented graphically in Figure 7:

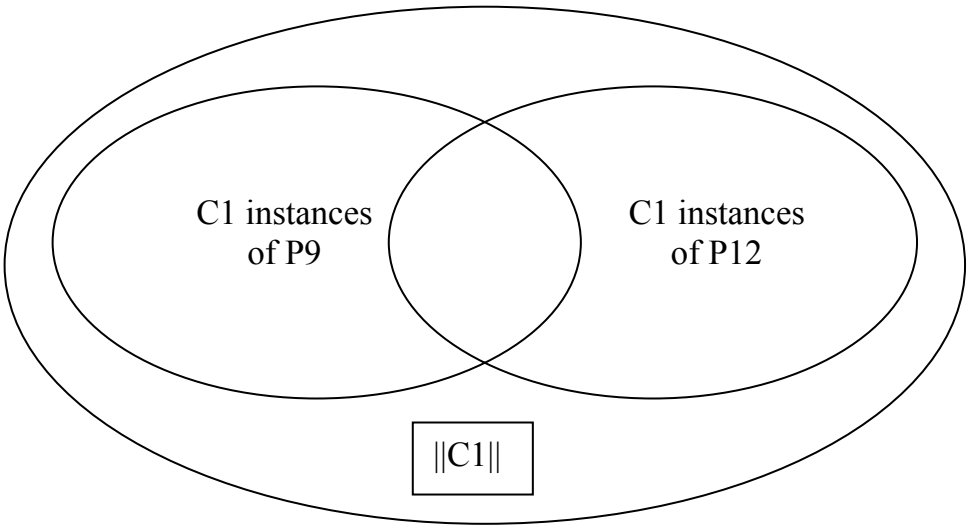


Figure 7 : A common class between two peers

The maximum number of times that a C1 instance in the PDMS appears in P9 is $|C1|_{P9}$ and the maximum number of times a C1 instance in the PDMS appears in P12 is $|C1|_{P12}$. The probabilities p, q that a specific C1 instance of the PDMS appears in P9, P12 respectively is

$p_{P9}(C1) = \frac{|C1|_{P9}}{||C1||}$ ***(3.1)***

$q_{P12}(C1) = \frac{|C1|_{P12}}{||C1||}$

As a result, taking into account formula (3.1), the probability of the common C1 instances of P9 and P12 in the independence overlap case, is

$\text{overlap}_{P9,P12,P14}(C1) = p_{P9}(C1) * q_{P12}(C1) * r_{P14}(C1)$ ***(3.2)***

Formula (3.2) expresses overlap as a probability. The overlap cardinality of peers P9 and P12 with respect to C1 is

$\text{Overlap}_{P9,P12}(C1) = \text{overlap}_{P9,P12}(C1) * ||C1||$ ***(3.3)***

As a result, the cardinality of the union of these peers with respect to class C1 is expressed as

$|C1|_{P9,P12} = |C1|_{P9} + |C1|_{P12} - \text{Overlap}_{P9,P12}(C1)$ ***(3.4)***

Then for the different cases of overlap, the estimation formulae are presented in Table 3.2 :

Overlap Cases	**Overlap Estimation**
Disjointness	$\text{overlap}_{P9,P12}(C1) = 0$
Independence	$\text{overlap}_{P9,P12}(C1) = \frac{\|C1\|_{P9}}{\|\|C1\|\|} * \frac{\|C1\|_{P12}}{\|\|C1\|\|}$
Quantified Overlap	$\text{overlap}_{P9,P12}(C1) = X$ where X is a known probability
Containment (e.g. of $C1_{P9}$ in $C1_{P12}$)	$\text{overlap}_{P9,P12}(C1) = 1$ $\text{overlap}_{P12,P9}(C1) = \frac{\|C1\|_{P9}}{\|C1\|_{P12}}$

Table 3.2 : Overlap cases for two peers exporting the same class

The case of disjointness is obvious : there are no common instances between P9 and P12 with respect to C1 and as a result $\text{overlap}_{P9,P12}(C1) = 0$. In the case of independence, $\text{overlap}_{P9,P12}(C1)$ is formed by the probabilities that a random C1 instance of the reference set, denoted as ||C1||, appears in the C1 instances of P9 and P12 respectively. When both of the peers contain a great percentage of the C1 instances of the PDMS, it is obvious that there is a greater probability that the C1 instances of the peers overlap. When one or both of the peers contain a small percentage

of the C1 instances of the PDMS, there is a small probability that their contents with respect to C1 overlap. When we consider quantified overlap of C1 in peers P9 and P12 we consider a probability that these peers have common C1 instances, e.g. 20% of their C1 instances may be common. In the case of containment overlap of $C1_{P9}$ in $C1_{P12}$, the probability of the C1 instances of P1 that are contained in P12, is $overlap_{P9,P12}(C1) = 1$, since all the C1 instances of P9 are contained in P12. However, the probability that the C1 instances of P12 are contained in P9 cannot be estimated as in the case of independence, where we assume a normal distribution of the C1 instances among the peers. It is $overlap_{P12,P9}(C1) = \frac{|C1|_{P9}}{|C1|_{P12}}$. In this case, the greater the number of C1 instances contained in P9, the greater the $overlap_{P12,P9}(C1)$.

3.2 PEERS PUBLISHING SUBSUMED CLASSES

The case where one peer exports instances of a class and another peer exports instances of a subclass is similar. In Figure 3, peer P12 exports instances of class C2 and peer P13 exports instances of class C6, which is a subclass of C2. In this case, we are interested in the probability of the common instances between the C2 instances in peer P12 and the C6 instances of peer P13, as shown graphically in Figure 8.

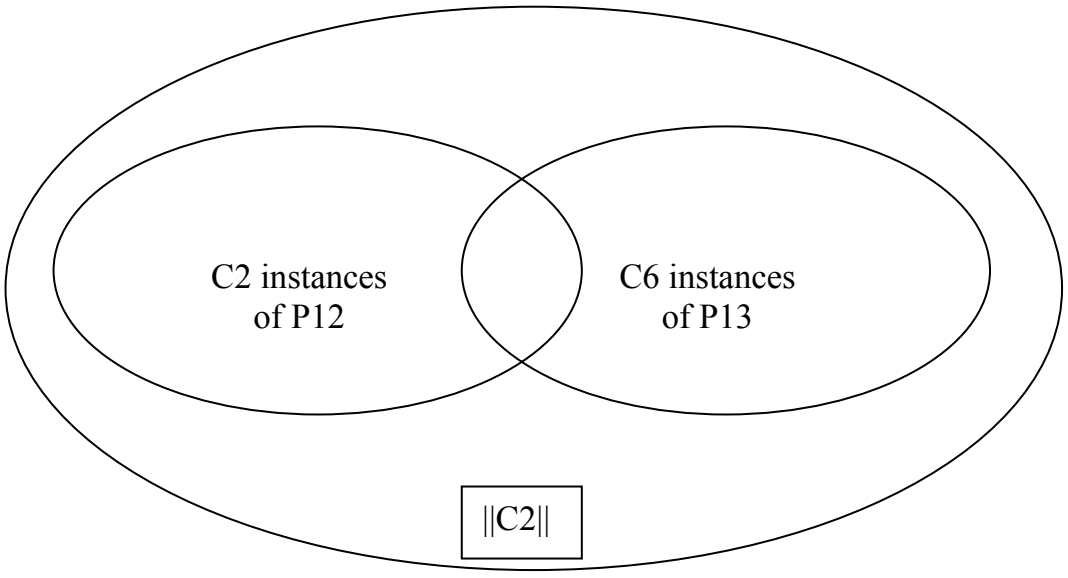


Figure 8 : Common class/subclass instances between two peers

In other words, we are interested in the probability that a C6 instance of P13 is also exported as a C2 instance of P12. In this case, the reference set for overlap estimations is ||C2||, because all

the instances of C6 can also be exported as instances of C2, due to the fact that C6 is a subclass of C2.

The probability p that a specific C2 instance of the PDMS appears in P12 is

$$p_{P12}(C2) = \frac{|C2|_{P12}}{||C2||}$$

and the probability that a C6 instance appears in P13 is

$$q_{P13}(C6) = \frac{|C6|_{P13}}{||C2||}$$

The formula for their overlap in the independence case is estimated as

$$overlap_{P124,P13}(C2UC6) = p_{P12}(C2) * q_{P13}(C6) \quad \textbf{(3.5)}$$

Formula (3.5) expresses overlap as a probability. The overlap cardinality of peers P12 and P13 with respect to C2UC6 is

$$Overlap_{P12,P13}(C2UC6) = overlap_{P12,P13}(C2UC6) * ||C2|| \quad \textbf{(3.6)}$$

As a result, the cardinality of the union of these peers with respect to class C6 is expressed as

$$|C2UC6|_{P12,P13} = |C2|_{P12} + |C6|_{P13} - Overlap_{P12,P13}(C2UC6) \quad \textbf{(3.7)}$$

Then for the different cases of overlap, the overlap estimation formulae are presented in Table 3.3 :

Overlap Cases	**Overlap Estimation**
Disjointness	$overlap_{P12,P13}(C2UC6) = 0$
Independence	$overlap_{P12,P13}(C2UC6) = \frac{\|C2\|_{P12}}{\|\|C2\|\|} * \frac{\|C6\|_{P13}}{\|\|C2\|\|}$
Quantified Overlap	$overlap_{P12,P13}(C2UC6)$ = X where X is a known probability
Containment (of $C6_{P13}$ in $C2_{P12}$)	$overlap_{P13,P12}(C2UC6) = 1$ $overlap_{P12,P13}(C2UC6) = \frac{\|C6\|_{P13}}{\|C2\|_{P12}}$

Table 3.3 : Overlap cases for two peers exporting subsumed classes

The case of disjointness is obvious: there are no common C6 instances between P12 and P13 and as a result $overlap_{P12,P13}(C2UC6) = 0$. In the case of independence, the probability of their

common instances is the probability that a C6 instance of P13, expressed by the probability that a C6 instance of the PDMS appears in P13, is exported as a C2 instance of P12, expressed by the probability that a C2 instance appears in P12. When there is a great probability that a C6 instance appears in P13 and a C2 instance appears in P12, there is also a great probability that there exist C6 instances of P13 that are exported also as C2 instances of P12, i.e. the overlap estimation is greater. Overlap is small when a small percentage of the C6 instances of the PDMS appear in P13 and also a small percentage of the C2 instances of the PDMS appear in P12. The quantified overlap of C2 and C6 in peers P12 and P13 is the probability that these peers have common instances, e.g. 20% of the C6 instances of P13 may also be exported as C2 instances of P12. In the case of containment overlap of $C6_{P13}$ in $C2_{P12}$, the probability of the C6 instances of P13 that are contained in P12, is $overlap_{P13,P12}(C2UC6) = 1$, since all the C6 instances of P13 are contained in P12. However, the probability of the C2 instances of P12 that are contained in P13 (of course the common instances will be instances of class C6), cannot be estimated assuming a normal distribution of the C6 instances of the PDMS. It is $overlap_{P12,P13}(C2UC6) = \frac{|C6|_{P13}}{|C2|_{P12}}$. All these C6 instances are also C2 instances of P12. In this case, the greater the probability that a C6 instance appears in P13, the greater the overlap of C2 and C6 in peers P12, P13.

3.3 PEERS PUBLISHING THE SAME PROPERTY

When two or more peers export the same property, one of the following cases holds :

- The common property has the same domain and range classes in both peers.
- One or both of the domain/range classes of the property in the one peer are subclasses of the corresponding domain/range classes of the property in the other peer.

We should point out that in the case of a property p relating two classes C and C', the reference set for the overlap estimation formulae is the Cartesian product of the instances of C and C' in the PDMS, whether we consider the overlap of peers that export the same property, or the overlap of peers that export subsumed properties with the same domain and range classes. This assumption also holds for more complex fragments that consist of two or more properties. When we have peers that export property p with domain class C and range class C', their overlap with respect to p is the probability that their common C and C' instances are connected to form p instances. In the following sections we will present the formulae for each case.

3.3.1 PROPERTIES WITH THE SAME DOMAIN/RANGE

Another case of fragment that two or more peers may export is a single property. In Figure 3, peers P9 and P12 export property r1, with domain class C1 and range class C2. In this case, the reference set for overlap estimation is the maximum possible number of r1 inatances in the PDMS, which is expressed by the Cartesian product of the classes that property r1 relates, i.e.||C1|| * ||C2||.

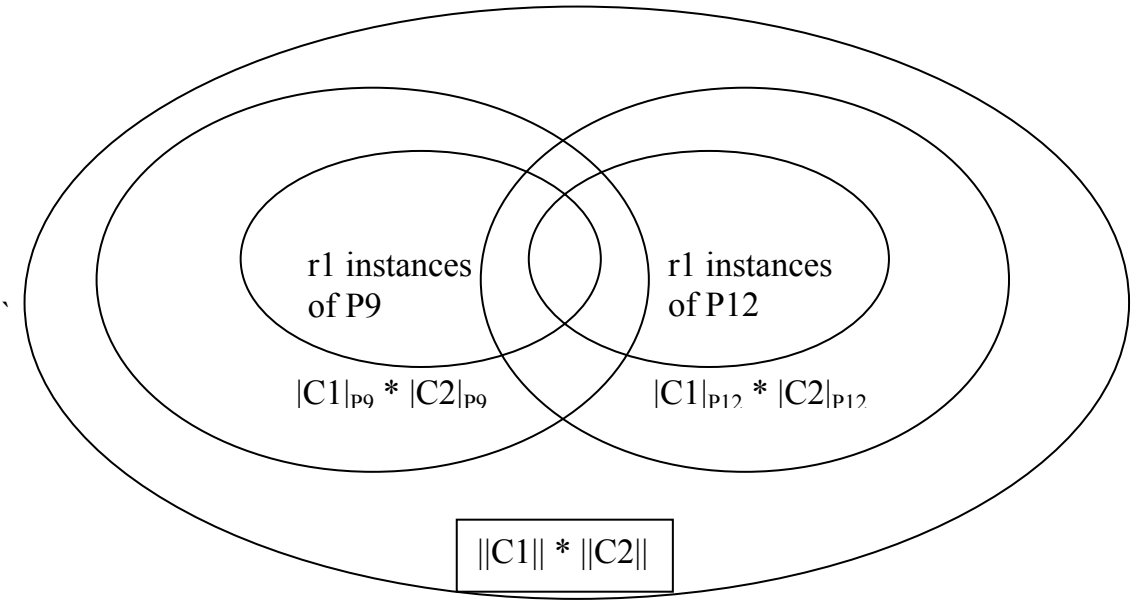


Figure 9 : A common property between two peers

We represent as $|C1|_{P9}$, $|C2|_{P9}$, $|C1|_{P12}$, $|C2|_{P12}$ the cardinalities of C1 and C2 in P9, P12 respectively. We also represent as $|r1|_{P9}$ and $|r1|_{P12}$ the cardinalities of r1 in P9, P12 respectively. To estimate the overlap between P9 and P12 with respect to r1 in the independence overlap case we would like to find the probability that a random instance of property r1 of the reference set is contained in both peers.

The probability that an r1 instance appears in peer P9 is

$$pr_{P9}(r1) = \frac{|r1|_{P9}}{||C1|| * ||C2||} \quad \textbf{(3.8)}$$

The probability of r1 in P12 is computed similarly to (3.8). So, we have :

$$overlap_{P9,P12}(r1) = pr_{P9}(r1) * pr_{P12}(r1) \quad \textbf{(3.9)}$$

Formula (3.9) expresses overlap as a probability. The overlap cardinality of peers P9 and P12 with respect to property r1 is

$$\text{Overlap}_{P9,P12}(r1) = \text{overlap}_{P9,P12}(r1) * ||C1|| * ||C2|| \quad \textbf{(3.10)}$$

As a result, the cardinality of the union of these peers with respect to property r1 is expressed as

$$|r1|_{P9,P12} = |r1|_{P9} + |r1|_{P12} - \text{Overlap}_{P9,P12}(r1) \quad \textbf{(3.11)}$$

There is also another, equivalent way to estimate the probability of property r1 in peers P9 and P12. The probability that the C1 and C2 instances of P9 are connected to form r1 instances is

$$p_{P9}(r1) = \frac{|r1|_{P9}}{|C1|_{P9} * |C2|_{P9}} \quad \textbf{(3.12)}$$

In the same way we can compute the probability that the C1 and C2 instances of P12 are connected to form r1 instances. In other words, the overlap of peers P9 and P12 with respect to property r1 is the product of the probabilities of their common C1 and C2 instances, using formula (3.2) and the probabilities that their common C1 and C2 instances form instances of property r1, using formula (3.12) :

$$\text{overlap}_{P9,P12}(r1) = \text{overlap}_{P9,P12}(C1) * \text{overlap}_{P9,P12}(C2) * pr_{P9}(r1) * pr_{P12}(r1) \quad \textbf{(3.13)}$$

In the following, we consider the overlap cases for C1 and C2. To be completely correct, we should take the case where e.g. C1 is independent in peers P9, P12 while C2 has quantified overlap in both peers and the case where C2 is independent in peers P9, P12 while C1 has quantified overlap in both peers. However, in favor of simplicity we consider only one of these cases, as the other is similar. This is done in all overlap estimation formulae presented in the following sections.

Then, for the different cases the overlap estimation formulae are presented in Table 3.4:

Overlap of C1 and C2 in peers P9, P12	**Overlap Estimation Formulae for C1, C2**	**Overlap Estimation Formulae for property r1**
Disjointness of C1, C2	$\text{overlap}_{P9,P12}(C1) = 0$ $\text{overlap}_{P9,P12}(C2) = 0$	$\text{overlap}_{P9,P12}(r1) = 0$

Disjointness of C1 and independence of C2	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = \frac{\|C2\|_{P9}}{\|C2\|} * \frac{\|C2\|_{P12}}{\|C2\|}$	$overlap_{P9,P12}(r1) = 0$
Disjointness of C1 and quantified overlap of C2	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = X$, where X is a known probability	$overlap_{P9,P12}(r1) = 0$
Disjointness of C1 and containment of $C2_{P12}$ in $C2_{P9}$	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = \frac{\|C2\|_{P12}}{\|C2\|_{P9}}$ $overlap_{P12,P9}(C2) = 1$	$overlap_{P9,P12}(r1) = 0$
Independence of C1, C2	$overlap_{P9,P12}(C1) = \frac{\|C1\|_{P9}}{\|C1\|} * \frac{\|C1\|_{P12}}{\|C1\|}$ $overlap_{P9,P12}(C2) = \frac{\|C2\|_{P9}}{\|C2\|} * \frac{\|C2\|_{P12}}{\|C2\|}$	$overlap_{P9,P12}(r1) = \frac{\|r1\|_{P9}}{\|C1\|*\|C2\|} * \frac{\|r1\|_{P12}}{\|C1\|*\|C2\|}$
Containment of $C1_{P12}$ in $C1_{P9}$ and containment of $C2_{P12}$ in $C2_{P9}$	$overlap_{P9,P12}(C1) = \frac{\|C1\|_{P12}}{\|C1\|_{P9}}$ $overlap_{P12,P9}(C1) = 1$ $overlap_{P9,P12}(C2) = \frac{\|C2\|_{P12}}{\|C2\|_{P9}}$ $overlap_{P12,P9}(C2) = 1$	$overlap_{P12,P9}(r1) = 1$ $overlap_{P9,P12}(r1) = \frac{\|r1\|_{P12}}{\|C1\|*\|C2\|}$

Table 3.4 : Overlap cases for two peers exporting the same property with the same domain/range

As shown in Table 3.4, all the cases where at least one of C1, C2 is disjoint in peers P9, P12, result in disjointness of r1 in these peers, i.e. $overlap_{P9,P12}(r1) = 0$. This is obvious, since only common C1 and C2 instances in the two peers can form common r1 instances. In the case of independence overlap of C1 and C2 in the two peers, $overlap_{P9,P12}(r1)$ is the probability that a random r1 instance of the PDMS appears in both peers. In the case where the C1 and C2 instances of P12 are contained in P9, $overlap_{P9,P12}(r1)$ is the probability that a random C1 instance

and a random C2 instance of the PDMS appear in P12, since peer P9 has C1 and C2 instances other than their common, while $overlap_{P12,P9}(r1) = 1$, since all the C1 and C2 instances of peer P12 are contained in P9. The probabilities of the common C1 and C2 instances between P9 and P12 are formed in the same way as in section 3.2. It is clear that when each of the peers P9 and P12 contain a great percentage of C1 and C2 instances, then the probabilities of their common C1 and C2 instances will also be great and if there is a great percentage of the C1 and C2 instances in each peer that connect to form r1 instances, then the probability of the overlap of property r1 in the two peers will also be great.

3.3.2 PROPERTIES WITH SUBSUMED DOMAINS/RANGES

In Figure 10, peer P1 exports property r1 with domain class C1 and range class C2, while peer P4 exports property r1 with domain class C5 and range class C6, which are subclasses of C1 and C2 respectively. The reference set for the overlap estimation formulae is the Cartesian product of instances of C1 and C2 in the PDMS, i.e. ||C1|| * ||C2||, since C5 and C6 are subclasses of C1 and C2 respectively and thus all the instances of C5 and C6 are contained in ||C1|| and ||C2|| respectively. The overlap of P1 and P4 with respect to r1, denoted as $overlap_{P1,P4}(r1)$, in the independence overlap case, is the probability that a random instance of property r1 belongs both to peers P1 and P4.

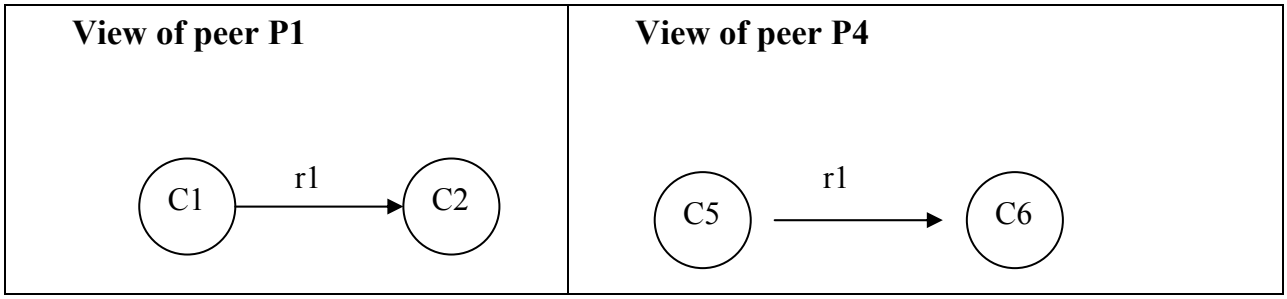


Figure 10 : Peers exporting the same property with subsumed domain and range

Then, for the different cases of r1 overlap the overlap estimation formulae are presented in Table 3.5:

Overlap of C1, C5 and C2, C6 in peers P1, P4	**Overlap Estimation Formulae for C1, C5 and C2, C6**	**Overlap Estimation Formulae for property r1**

Disjointness of C1,C5 and C2, C6	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = 0$	$overlap_{P1,P4}(r1) = 0$
Disjointness of C1, C5 and independence of C2, C6	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = \frac{\|C2\|_{P1}}{\|C2\|} * \frac{\|C6\|_{P4}}{\|C2\|}$	$overlap_{P1,P4}(r1) = 0$
Disjointness of C1, C5 and quantified overlap of C2, C6	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6)$ = X, where X is a known probability	$overlap_{P1,P4}(r1) = 0$
Disjointness of C1, C5 and containment of $C6_{P4}$ in $C2_{P1}$	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = \frac{\|C6\|_{P4}}{\|C2\|_{P1}}$ $overlap_{P4,P1}(C2UC6) = 1$	$overlap_{P1,P4}(r1) = 0$
Independence of C1, C5 and C2, C6	$overlap_{P1,P4}(C1UC5) = \frac{\|C1\|_{P1}}{\|C1\|} * \frac{\|C5\|_{P4}}{\|C1\|}$ $overlap_{P1,P4}(C2UC6) = \frac{\|C2\|_{P1}}{\|C2\|} * \frac{\|C6\|_{P4}}{\|C2\|}$	$overlap_{P1,P4}(r1) = \frac{\|r1\|_{P1}}{\|C1\|*\|C2\|} * \frac{\|r1\|_{P4}}{\|C1\|*\|C2\|}$
Containment of $C5_{P4}$ in $C1_{P1}$ and containment of $C6_{P4}$ in $C2_{P1}$	$overlap_{P1,P4}(C1UC5) = \frac{\|C5\|_{P4}}{\|C1\|_{P1}}$ $overlap_{P4,P1}(C1UC5) = 1$ $overlap_{P1,P4}(C2UC6) = \frac{\|C6\|_{P4}}{\|C2\|_{P1}}$ $overlap_{P4,P1}(C2UC6) = 1$	$overlap_{P4,P1}(r1) = 1$ $overlap_{P1,P4}(r1) = \frac{\|r1\|_{P4}}{\|C1\|*\|C2\|}$

Table 3.5 : Overlap cases for two peers exporting the same property with subsumed domains/ranges

As shown in Table 3.5, the overlap cases are the same as those of Table 3.4 and the overlap formulae are similar. The only difference is that we consider subsumed classes (i.e. classes C1 and C5 and classes C2 and C6). It is clear that when each of the peers P1 and P4 contain a great

percentage of C5 and C6 instances, then the probabilities of their common C5 and C6 instances will also be great as well as the probability of the overlap of property r1 in the two peers.

The overlap of Table 3.5 can be expressed as cardinality with the following formula :

$Overlap_{P1,P4}(r1) = overlap_{P1,P4}(r1) * ||C1|| * ||C2||$

As a result, the cardinality of the union of these peers with respect to property r1 is expressed as

$|r1|_{P1,P4} = |r1|_{P1} + |r1|_{P4} - Overlap_{P1,P4}(r1)$

3.4 PEERS PUBLISHING SUBSUMED PROPERTIES

When two or more peers export the same property, one of the following cases holds :

- The properties have the same domain and range classes in both peers.
- One or both of the domain/range classes of the property in the one peer are subclasses of the corresponding domain/range classes of the property in the other peer.

In the following sections we will present the formulae for each case.

3.4.1 PROPERTIES WITH THE SAME DOMAIN/RANGE

In Figure 11 peer P1 exports property r1 and peer P3 exports property r4, subproperty of r1 with the same domain and range.

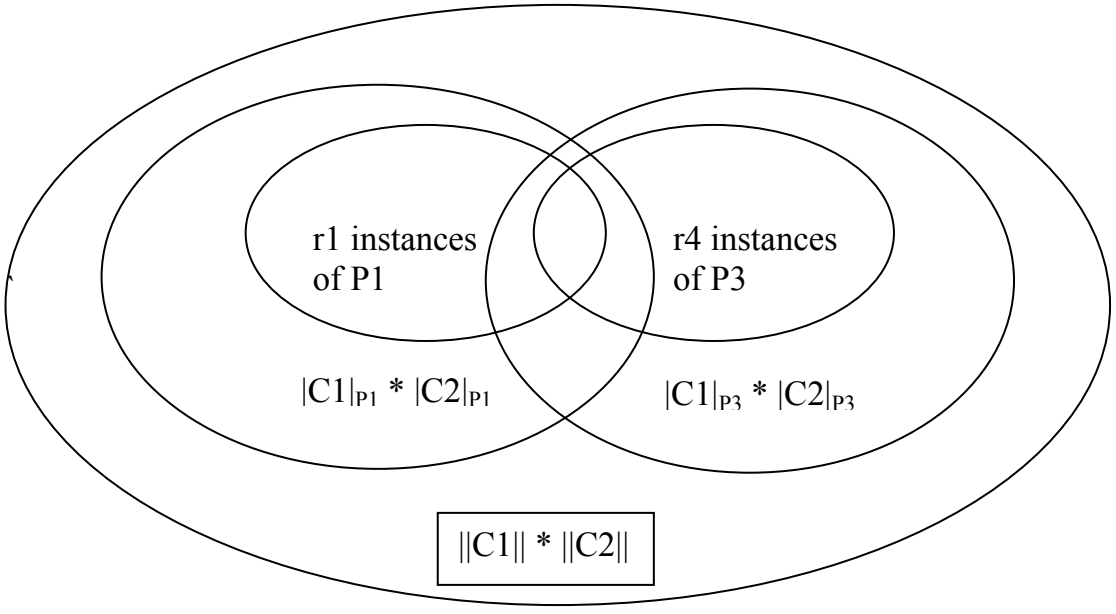


Figure 11 : Peers exporting subsumed properties with the same domain / range

The reference set for the overlap estimation formulae is the Cartesian product of instances of C1 and C2 in the PDMS, i.e. ||C1|| * ||C2||. The overlap of P1 and P3 with respect to r4, denoted as $overlap_{P1,P3}$(r1Ur4) in the independence case, is the probability that a random instance of property r4 of the reference set belongs both to peers P1 and P3. Then, for the different cases of r1 and r4 overlap the overlap estimation formulae are presented in Table 3.6:

Overlap of C1 and C2 in peers P1, P3	**Overlap Estimation Formulae for C1 and C2**	**Overlap Estimation Formulae for properties r1, r4**
Disjointness of C1 and C2	$overlap_{P1,P3}(C1) = 0$ $overlap_{P1,P3}(C2) = 0$	$overlap_{P1,P3}(r1Ur4) = 0$
Disjointness of C1 and independence of C2	$overlap_{P1,P3}(C1) = 0$ $overlap_{P1,P3}(C2)=\frac{\|C2\|_{P1}}{\|C2\|} * \frac{\|C2\|_{P3}}{\|C2\|}$	$overlap_{P1,P3}(r1Ur4) = 0$
Disjointness of C1 and quantified overlap of C2	$overlap_{P1,P3}(C1) = 0$ $overlap_{P1,P3}(C2)$ = X, where X is a known probability	$overlap_{P1,P3}(r1Ur4) = 0$
Disjointness of C1 and containment of $C2_{P3}$ in $C2_{P1}$	$overlap_{P1,P3}(C1) = 0$ $overlap_{P1,P3}(C2) = \frac{\|C2\|_{P3}}{\|C2\|_{P1}}$ $overlap_{P3,P1}(C2) = 1$	$overlap_{P1,P3}(r1Ur4) = 0$
Independence of C1 and C2	$overlap_{P1,P3}(C1)=\frac{\|C1\|_{P1}}{\|C1\|} * \frac{\|C1\|_{P3}}{\|C1\|}$ $overlap_{P1,P3}(C2)=\frac{\|C2\|_{P1}}{\|C2\|} * \frac{\|C2\|_{P3}}{\|C2\|}$	$overlap_{P1,P3}(r1Ur4)=$ $\frac{\|r1\|_{P1}}{\|C1\|*\|C2\|} * \frac{\|r4\|_{P3}}{\|C1\|*\|C2\|}$
Containment of $C1_{P3}$ in $C1_{P1}$ and containment of $C2_{P3}$ in $C2_{P1}$	$overlap_{P1,P3}(C1) = \frac{\|C1\|_{P3}}{\|C1\|_{P1}}$ $overlap_{P3,P1}(C1) = 1$ $overlap_{P1,P3}(C2) = \frac{\|C2\|_{P3}}{\|C2\|_{P1}}$ $overlap_{P3,P1}(C2) = 1$	$overlap_{P1,P3}(r1Ur4)=$ $\frac{\|r4\|_{P3}}{\|C1\|*\|C2\|}$ $overlap_{P3,P1}(r1Ur4) = 1$

Table 3.6 : Overlap cases for two peers exporting subsumed properties with the same domain/range

As shown in Table 3.6, the overlap cases are completely the same as those in Table 3.4. and Table 3.5. Since we consider the same domain and range classes C1 and C2 in the subsumed properties r1 and r4, the overlap formulae for C1 and C2 in peers P1 and P3 are the same as the respective formulae of Table 3.4. The only difference is that since we have subsumed properties, the $overlap_{P1,P3}(r1Ur4)$ will involve the probabilities of the common C1 and C2 instances between P1 and P3 and the probability that a random r4 instance of the PDMS appears both in peers P1 and P3.

The overlap of Table 3.6 can be expressed as cardinality with the following formula :

$Overlap_{P1,P3}(r1Ur4) = overlap_{P1,P3}(r1Ur4) * ||C1|| * ||C2||$

As a result, the cardinality of the union of these peers with respect to r1Ur4 is expressed as

$|r1Ur4|_{P1,P3} = |r1|_{P1} + |r4|_{P3} - Overlap_{P1,P3}(r1Ur4)$

3.4.2 PROPERTIES WITH SUBSUMED DOMAINS/RANGES

Another case of two peers exporting subsumed properties is that where only the domain/range class of the subproperty may be the same as that of the parent property, while the range/domain class may be a subclass of the range/domain class of the parent property, e.g. in Figure 12, peer P6 exports r4 and peer P13, exports r4 with the same range, class C6, but class C7 as domain class, which is a subclass of C5.

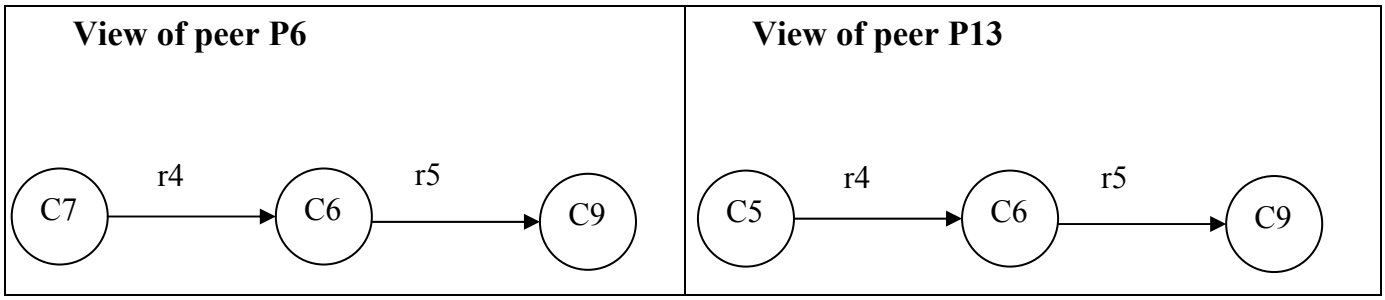


Figure 12 : Peers exporting subsumed properties with subsumed domain / range

However, there is also the case where both the domain and range classes of the subproperty may be subclasses of the domain and range classes of the parent property, e.g. peer P9 of Figure 3 exports r1 and peer P13 exports r4, while classes C7 and C6 are subclasses of C1 and C2 respectively. This is graphically presented in Figure 13:

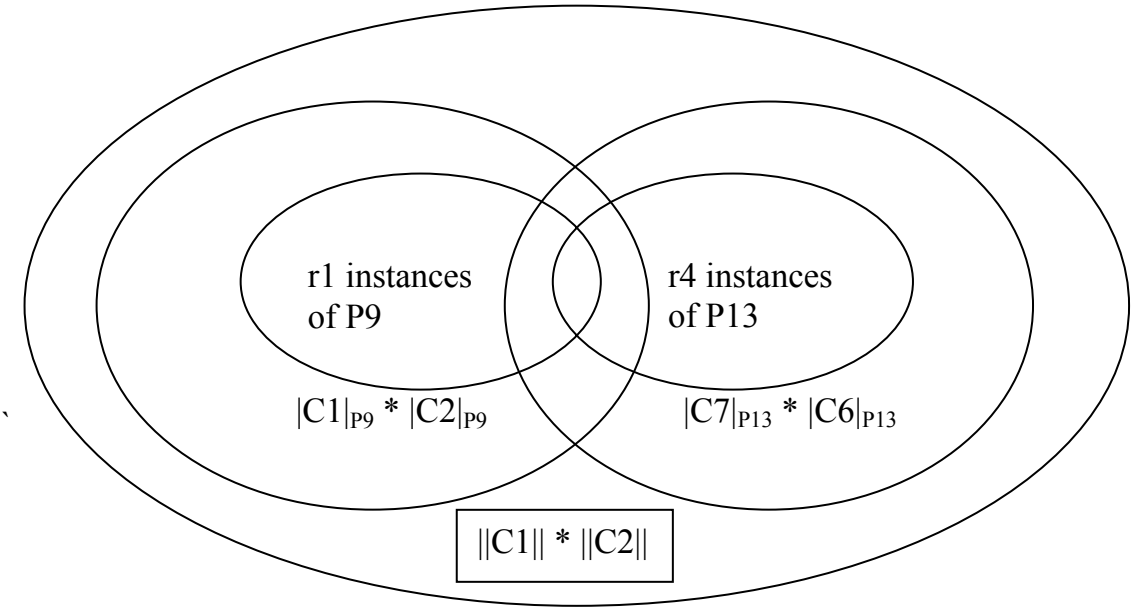


Figure 13 : Common property/subproperty instances between two peers

The overlap of peers P9 and P13 with respect to properties r1 and r4 in the independence overlap case will involve the probability that a random r4 instance of the PDMS appears in P13 and at the same time it is exported as an r1 instance in peer P9. The overlap expressed as probability is similar to (3.9), using formula (3.8) :

$overlap_{P9,P13}(r1Ur4) = pr_{P9}(r1) * pr_{P13}(r4)$

The overlap cardinality of peers P9 and P13 with respect to r1Ur4 is

$Overlap_{P9,P13}(r1Ur4) = overlap_{P9,P13}(r1Ur4) * ||C1|| * ||C2||$

As a result, the cardinality of the union of these peers with respect to property r4 is expressed as

$|r1Ur4|_{P9,P3} = |r1|_{P9} + |r4|_{P13} - Overlap_{P9,P13}(r1Ur4)$

The equivalent overlap formula is shown below, taking formula (3.13) into account :

$$overlap_{P9,P13}(r1Ur4) = overlap_{P9,P13}(C1UC5)*overlap_{P9,P13}(C2UC6) * pr_{P9}(r1) * pr_{P13}(r4) \quad \textbf{(3.14)}$$

Then, for the different cases of r1 and r4 overlap the overlap estimation formulae are presented in Table 3.7. The overlap cases of Table 3.7 between domain and range classes are the same as those of Table 3.6. The only difference in the overlap formulae is that since we consider

subsumed domain/range classes, we try to estimate the probability of common instances between e.g. the domain/range class of P9 and its subsumed class, which is the domain/range class of P13.

It is clear that when each of the peers P9 and P13 contain a great percentage of C7 and C6 instances, then the probabilities of their common C7 and C6 instances will also be great, as well as the probability of the overlap of properties r1, r4 in the two peers.

Overlap of C1, C7 and C2, C6 in peers P9, P13	**Overlap Estimation Formulae for C1, C7 and C2, C6**	**Overlap Estimation Formulae for properties r1, r4**
Disjointness of C1 C7 and C2, C6	$overlap_{P9,P13}(C1 \cup C7) = 0$ $overlap_{P9,P13}(C2 \cup C6) = 0$	$overlap_{P9,P13}(r1 \cup r4) = 0$
Disjointness of C1, C7 and independence of C2, C6	$overlap_{P9,P13}(C1 \cup C7) = 0$ $overlap_{P9,P13}(C2 \cup C6) = \frac{\|C2\|_{P9}}{\|C2\|} * \frac{\|C6\|_{P13}}{\|C2\|}$	$overlap_{P9,P13}(r1 \cup r4) = 0$
Disjointness of C1, C7 and quantified overlap of C2, C6	$overlap_{P9,P13}(C1 \cup C7) = 0$ $overlap_{P9,P13}(C2 \cup C6) = X$, where X is a known probability	$overlap_{P9,P13}(r1 \cup r4) = 0$
Disjointness of C1, C7 and containment of $C6_{P13}$ in $C2_{P9}$	$overlap_{P9,P13}(C1 \cup C7) = 0$ $overlap_{P9,P13}(C2 \cup C6) = \frac{\|C6\|_{P13}}{\|C2\|_{P9}}$ $overlap_{P13,P9}(C2 \cup C6) = 1$	$overlap_{P9,P13}(r1 \cup r4) = 0$
Independence of C1, C7 and C2, C6	$overlap_{P9,P13}(C1 \cup C7) = \frac{\|C1\|_{P9}}{\|C1\|} * \frac{\|C7\|_{P13}}{\|C1\|}$ $overlap_{P9,P13}(C2 \cup C6) = \frac{\|C2\|_{P9}}{\|C2\|} * \frac{\|C6\|_{P13}}{\|C2\|}$	$overlap_{P9,P13}(r1 \cup r4) = \frac{\|r1\|_{P9}}{\|C1\| * \|C2\|} * \frac{\|r4\|_{P13}}{\|C1\| * \|C2\|}$

Containment of $C7_{P13}$ in $C1_{P9}$ and containment of $C6_{P13}$ in $C2_{P9}$	$overlap_{P9,P13}(C1UC7) = \frac{\|C7\|_{P13}}{\|C1\|_{P9}}$ $overlap_{P13,P9}(C1UC7) = 1$ $overlap_{P9,P13}(C2UC6) = \frac{\|C6\|_{P13}}{\|C2\|_{P9}}$ $overlap_{P13,P9}(C2UC6) = 1$	$overlap_{P9,P13}(r1Ur4) = \frac{\|r4\|_{P13}}{\|C1\| * \|C2\|}$

Table 3.7 : Overlap cases for two peers exporting subsumed properties with subsumed domains/ranges

3.5 OVERLAP FOR MORE THAN TWO PEERS

When there are more than two peers that export the same class/property or subsumed classes/properties we will make the simplifying assumption that we only have independence overlap among their data. Apart from the simplicity that this assumption offers, it is the only realistic assumption that we can make in a PDMS, where we do not (or we cannot) keep a detailed knowledge about the data of each peer.

Let us assume that we have peers P1, P2, … , PN. Let us denote as C the common class in the case that they export the same class and let C' be a subclass of C. Let us also denote as r the property in the case where they export the same property and let r' be a subproperty of r. We assume that class C1 is the domain class of property r and class C2 is the range class of r. In addition, we consider class C1' to be a subclass of C1 and class C2' to be a subclass of C2. In the case of a class C and its subclass C', the reference set for the overlap computations is ||C||, since all the instances of the subclass C' in the PDMS are involved in ||C||. In the case of a property r and a subproperty r' of it, the reference set for overlap computations is ||C1|| * ||C2||, since all the instances of the subproperty are contained in ||C1|| * ||C2||.

Table 3.8 shows the overlap formulae for each of the previous cases when there are more than two peers, assuming independence among their data and taking into account the formulae presented in Tables 3.1 – 3.6. We can compute the cardinality of overlap for each case of Table 3.8 and thus the cardinality of the union of peers with respect to the specific fragment in each case. For example, the overlap cardinality of peers P1, P2, …, PN with respect to property r can be written as $Overlap_{P1,...PN}(r) = overlap_{P1,...PN}(r) * ||C1|| * ||C2||$

Same class	**Subsumed classes**
$overlap_{P1,\ldots PN}(C) = \frac{\|C\|_{P1}}{\|C\|} * \frac{\|C\|_{P2}}{\|C\|} *\ldots* \frac{\|C\|_{PN}}{\|C\|}$	$overlap_{P1,\ldots PN}(C \cup C') = \frac{\|C\|_{P1}}{\|C\|} * \frac{\|C'\|_{P2}}{\|C\|} * \ldots * \frac{\|C\|_{PN}}{\|C\|}$
Same property (same domain/range)	**Same property (subsumed domain/range)**
$overlap_{P1,\ldots PN}(r) = \frac{\|r\|_{P1}}{\|C1\|*\|C2\|} *\ldots* \frac{\|r\|_{PN}}{\|C1\|*\|C2\|}$	$overlap_{P1,\ldots PN}(r) = \frac{\|r\|_{P1}}{\|C1\|*\|C2\|} *\ldots* \frac{\|r\|_{PN}}{\|C1\|*\|C2\|}$
Subsumed properties (same domain/range)	**Subsumed properties (subsumed domain/range)**
$overlap_{P1,\ldots PN}(r \cup r') = \frac{\|r\|_{P1}}{\|C1\|*\|C2\|} *\ldots* \frac{\|r'\|_{PN}}{\|C1\|*\|C2\|}$	$overlap_{P1,\ldots PN}(r \cup r') = \frac{\|r\|_{P1}}{\|C1\|*\|C2\|} *\ldots* \frac{\|r'\|_{PN}}{\|C1\|*\|C2\|}$

Table 3.8 : Generalization of overlap among more than two peers

3.6 OVERLAP OF COMPLEX FRAGMENTS

Up to this point we only presented the overlap cases among peers with respect to a single class or a single property. However, the peers of a PDMS may export more complex fragments that involve two or more properties joined on a common class. For example, in Figure 3 we can see that peers P9 and P14 both export properties r1 and r3. The two properties are joined on C1. We need to estimate the common instances between P9 and P14 with respect to the join of r1 with r3, denoted as $overlap_{P9,P14}(r1 \bowtie r3)$. We should point out that the morphology of the join, discussed in detail in section 4.2.1, does not affect the number of instances of the join in a peer. In other words, the cardinality of the join is not affected by the join morphology. In our estimations we will not consider the possible overlap of single classes, in favor of simplicity.

The overlap of peers P9 and P14 with respect to $r1 \bowtie r3$ is the probability that a random instance of $r1 \bowtie r3$ in the PDMS is contained both in P9 and P14. The reference set for the overlap computations is the maximum possible number of $r1 \bowtie r3$ instances that can be retrieved in our PDMS. This corresponds to the product of the instances of all the classes which form the

fragment. In our case, the reference set is the product of all the instances of classes C1, C2 and C4 in the PDMS, i.e. ||C1|| * ||C2|| * ||C4||. To be completely correct, we should take all the overlap cases for each pair of classes. However, this is complicated and as a result we only consider independence overlap among classes. Then, for the overlap of r1 ⋈ r3 the estimation formula is presented in Table 3.9:

Overlap of C1, C2, C4 in peers P9, P14	**Overlap Estimation Formulae for C1, C2 and C4**	**Overlap Estimation Formula for fragment** r1 ⋈ r3
Independence of C1, C2 and C4	$overlap_{P9,P14}(C1)=\frac{\|C1\|_{P9}}{\|C1\|}*\frac{\|C1\|_{P14}}{\|C1\|}$ $overlap_{P9,P14}(C2)=\frac{\|C2\|_{P9}}{\|C2\|}*\frac{\|C2\|_{P14}}{\|C2\|}$ $overlap_{P9,P14}(C4)=\frac{\|C4\|_{P9}}{\|C4\|}*\frac{\|C4\|_{P14}}{\|C4\|}$	$overlap_{P9,P14}(r1 \bowtie r3)=$ $\frac{\|r1\times r3\|_{P9}}{\|C1\|*\|C2\|*\|C4\|}*\frac{\|r1\times r3\|_{P14}}{\|C1\|*\|C2\|*\|C4\|}$

Table 3.9 : Independence overlap of a complex fragment between two peers

If we denote the complex fragment r1 ⋈ r3 as F, then the formula of Table 3.9 for $overlap_{P9,P14}$(r1 ⋈ r3) can be written as

$$overlap_{P9,P14}(F) = |F|_{P9} * |F|_{P14} / \prod_k \|Ck\|^2 \quad \textbf{(3.15)}$$

where k = 1, 2, 4 and expressed as cardinality, it could be written as

$$Overlap_{P9,P14}(F) = |F|_{P9} * |F|_{P14} / \prod_k \|Ck\| \quad \textbf{(3.16)}$$

where k = 1, 2, 4

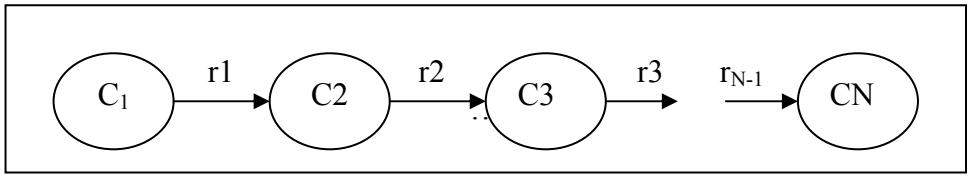


Figure 14 : Complex PDMS schema fragment example

Let us denote as F the fragment of Figure 14. Then, the overlap of the m peers with respect to this fragment, denoted as $overlap_{P1,P2...,Pm}(F)$ is the probability that a random instance of fragment F in the PDMS appears in all these peers. The reference set for overlap estimations in this case is the product of all the instances of the classes C1, C2, …, CN that form the fragment of Figure 14. Of course, since we consider more than two peers we show overlap estimations only for the case of independence among peers, in favor of simplicity. As a result, Table 3.10 is similar to Table 3.9, for the case of m peers all exporting a fragment of n-1 properties joined together. We should note that in the formulae of Table 3.10 where the symbol × appears, it denotes the join symbol ⋈ that we use :

Overlap of C1-CN in peers P1-Pm	**Overlap Estimation Formulae For C1 – CN**	**Overlap Estimation Formula for fragment F**
Independence of C1 – CN	$overlap_{P1,P2...,Pm}(C1)=\frac{\|C1\|_{P1}}{\|C1\|}*\frac{\|C1\|_{P2}}{\|C1\|}*\dots*\frac{\|C1\|_{Pm}}{\|C1\|}$ $overlap_{P1,P2...,Pm}(C2)=\frac{\|C2\|_{P1}}{\|C2\|}*\frac{\|C2\|_{P2}}{\|C2\|}*\dots*\frac{\|C2\|_{Pm}}{\|C2\|}$ ⋮ $overlap_{P1,P2...,Pm}(CN)=\frac{\|CN\|_{P1}}{\|CN\|}*\frac{\|CN\|_{P2}}{\|CN\|}*\dots*\frac{\|CN\|_{Pm}}{\|CN\|}$	$overlap_{P1,P2...,Pm}(F)=\frac{\|F\|_{P1}}{\|C1\|...*\|CN\|}*\dots*\frac{\|F\|_{Pm}}{\|C1\|*...*\|CN\|}$

Table 3.10 : Independence overlap of a complex fragment consisting of n-1 properties among m peers

As presented in Table 3.10, to estimate the overlap of a fragment comprising n-1 properties among m peers, we should first take overlap of every single class of the fragment among the peers. So, first of all we estimate $overlap_{P1,P2...,Pm}(Ci)$, $\forall\ 1 \le i \le N$. This means that we will estimate the probabilities of the common instances of all the classes among the m peers.

We can compute the cardinality of overlap for each case of Table 3.10 and thus the cardinality of the union of peers with respect to the specific fragment in each case. For example, the overlap cardinality of peers P1, P2, …, Pm with respect to fragment F can be written as

$Overlap_{P1,...Pm}(F) = overlap_{P1,...Pm}(F) * ||C1|| * ||C2|| * \ldots * ||CN||$ ***(3.17)***

Then, the cardinality of the union of P1, P2, .., Pm with respect to F is

$|F|_{P1,...Pm} = |F|_{P1} + \ldots + |F|_{Pm} - Overlap_{P1,...Pm}(F)$ ***(3.18)***

The overlap of Table 3.10 can also be written as

$overlap_{P1,...,Pm}(F) = pr_{P1}(F) * \ldots * pr_{Pm}(F) = |F|_{P1} / \prod_k ||Ck|| * \ldots * |F|_{Pm} / \prod_k ||Ck|| =$

$= \prod_i |F|_{Pi} / \prod_k ||Ck||^n$ ***(3.19)***

where i = 1, …, m and k = 1, … , N and expressed as cardinality, it could be written as

$Overlap_{P1,...,Pm}(F) = overlap_{P1,...,Pm}(F) * \prod_k ||Ck|| = \prod_i |F|_{Pi} / \prod_k ||Ck||^{n-1}$ ***(3.20)***

where i = 1, …, m and k = 1, … , N

3.7 CARDINALITY ESTIMATION OF A PDMS FRAGMENT FOR SEVERAL PEERS

In this section we are going to present how we can estimate the cardinality of an arbitrary fragment F in the PDMS, consisting of classes C1, C2, …, Cn that is exported by a specific number of peers.

To do this, let us assume that at a specific point in time there are only three peers in the PDMS that export fragment F, peers P1, P2 and P3. We will denote the cardinality of F in peer P as $|F|_P$. In addition, let us assume that peer P4 is the guide peer for fragment F. This means that every peer that joins the system and exports F publicizes its cardinality with respect to F to peer P4 and thus P4 keeps the total cardinality of F in the PDMS, denoted as $\sum_i |F|_{Pi}$. This sum contains also duplicate F instances, since there is also the overlap among the peers that is involved in $\sum_i |F|_{Pi}$.

In our example, the overlap of the PDMS peers that export F can be computed if we consider pairwise the common instances of the peers, as well as the common instances of all the peers of

the PDMS together. The relation between the cardinality of fragment instances and the various overlaps can be written as below :

$|F_{\{P1,\ P2,\ P3\}}| = |F|_{P1} + |F|_{P2} + |F|_{P3} - \text{Overlap}_{P1,P2}(F) - \text{Overlap}_{P1P3}(F) - \text{Overlap}_{P2P3}(F) + \text{Overlap}_{P1P2,P3}(F)$ ***(3.21)***

In formula by subtracting the pairwise overlap of peers with respect to F we eliminate completely the instances of F that are common in all peers. So, we should add the overlap of all peers in order to consider them once.

Figure 15 below illustrates our example. By considering $|F|_{P1}+|F|_{P2}+|F|_{P3}$, instances belonging e.g. to both P_1 and P_2 but not to P_3 (region ACD) are counted twice. The same is true for regions EBC and FBD, while instances belonging to P_1, P_2 and P_3 (region BCD) are counted three times. By eliminating all the overlaps between couples of peers (terms $\text{Overlap}_{P1,P2}(F)$, $\text{Overlap}_{P1P3}(F)$, $\text{Overlap}_{P2P3}(F)$), instances in the regions ACD, EBC and FBD are eliminated once (they were counted twice, so they remain counted once), but instances in BCD are eliminated three times (they were counted 3 times, so they are completely eliminated). By counting them once again (term $\text{Overlap}_{P1,P2,P3}(F)$) every instance in P_1, P_2 and P_3 is counted only once

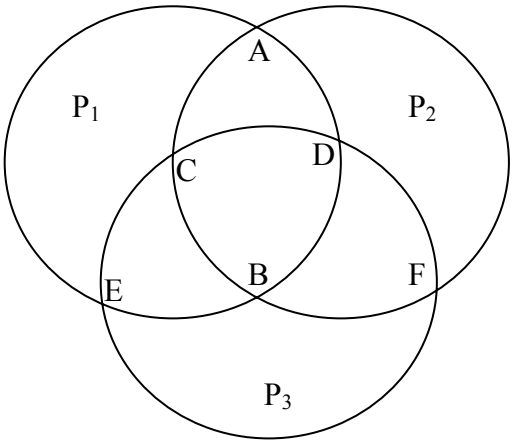


Figure 15 : Overlap of the instances of three peers

Likewise, if we consider that we have n peers in the PDMS that export F, we should first compute their overlap pairwise and subtract it, then compute the overlap of every three peers and subtract it etc. In the end we will need to add some overlaps that we have subtracted twice, in order to have them computed in the cardinality formula only once. Thus, for n peers that export fragment F we have

$|F|_{P1,\ldots,\ Pn} = \sum_i |F|_{Pi} - \sum_{ij} \text{Overlap}_{PiPj}(F) + \sum_{ijk} \text{Overlap}_{PiPjPk}(F) - \ldots + (-1)^{n-1} \text{Overlap}_{P1\ldots Pn}(F)$ ***(3.22)***

***Proof**: Any instance of F provided by one of the peers P_1, ..., P_n is characterized by the subset of peers where the instance can be found, e.g. P_{i1}, ..., P_{ik} (meaning that it is not present on the other peers). In Figure 14 above, this means that any instance belongs to only one of the disjoint regions partitioning the diagram. Let us note inst(SP) the set of instances found on the set of peers SP, but not on any other peer – this corresponds to one of the disjoint regions in the diagram.*

Notice that term Overlap $_{Pi1...Pik}$(F) contains inst({P_{i1}, ..., P_{ik}}), but also inst(SP) for any SP superset of {P_{i1}, ..., P_{ik}}. This means that instances of inst({P_{i1}, ..., P_{ik}}) appear in each Overlap$_{SP}$(F), where SP is a subset of {P_{i1}, ..., P_{ik}}. There are C_k^j subsets of {P_{i1}, ..., P_{ik}} of size j and from formula (7) the sign of an overlap of j peers is $(-1)^{j-1}$, so the number of times an instance appears is $C_k^1 - C_k^2 + ... + (-1)^{k-1}C_k^k$, which is always 1, because of the identity $\sum_{0\leq i\leq n} (-1)^i C_n^i = 0$.

In the next section we will provide formulae for overlap estimation between sets of peers that export the same fragment F. These formulae are needed for computing the overlaps in formula (3.23).

3.7.1 OVERLAP OVER SETS OF PEERS

Suppose that there are two sets of peers SP1 and SP2 that export the same fragment F, which comprises classes C1, C2, … , Ck and we are interested in computing their overlap with respect to F. In the case that these sets of peers are disjoint with each other, the overlap formulae are similar to formula (3.20) :

$$Overlap_{SP1,SP2}(F) = |F|_{SP1} * |F|_{SP2} / \prod_k ||C_k|| \quad \textbf{(3.23)}$$

To generalize, when we have n sets of peers SP1, SP2, … , SPn disjoint with each other, their overlap formula is

$$Overlap_{SP1,\ldots,SPn}(F) = \prod_i |F|_{SPi} / \prod_k ||C_k|| \quad \textbf{(3.24)}$$

Figure 16 shows the case where we have two sets of peers SP1 and SP2 and there is some overlap between them.The formula for the overlap computation of sets SP1 and SP2, is given below :

$$Overlap_{SP1,SP2}(F) = |F|_{SP1\cap SP2} + Overlap_{SP1-SP2,SP2-SP1}(F) - Overlap_{SP1\cap SP2,SP1-SP2,SP2-SP1}(F) \quad \textbf{(3.25)}$$

As formula (3.25) shows, in order to compute the overlap of two sets SP1 and SP2 we should take the cardinality of their intersection and then add the overlap with respect to F of those peers

that belong to set SP1 but do not belong to set SP2 (difference SP1 – SP2) and the overlap with respect to F of those peers that belong to SP2 but not to SP1 (difference SP2 – SP1). However, we should subtract the overlap with respect to F of peers that belong to SP1∩SP2, SP1 – SP2 and SP2 – SP1, since this overlap has already been counted twice.

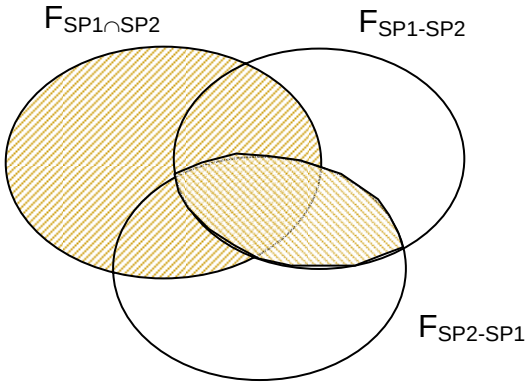


Figure 16 : Overlap between two sets of peers

As a generalization, Figure 17 shows graphically the case of n peer sets SP1, SP2, … , SPn that overlap with each other :

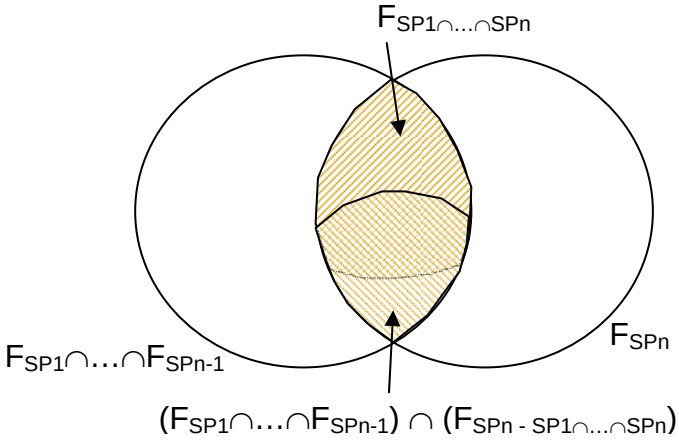


Figure 17 : Overlap among n sets of peers

In this case we have the intersection of sets SP1, … , SPn-1 and we would like to compute the overlap between this intersection and the set SPn. The formula for the overlap estimation among these sets of peers is given below :

$$Overlap_{SP1,\ldots,SPn}(F) \approx |F|_{SP1 \cap \ldots \cap SPn} + Overlap(F_{SP1 \cap \ldots \cap F_{SPn-1}}, F_{SPn - SP1 \cap \ldots \cap SPn}) - Overlap(F_{SP1 \cap \ldots \cap SPn}, F_{SPn - SP1 \cap \ldots \cap SPn}, F_{SP1 \cap \ldots \cap F_{SPn-1}}) \quad \textbf{(3.26)}$$

According to formula (3.26), the overlap of sets SP1, SP2, … , SPn with respect to F is approximately equal to the sum of the cardinality of the intersection of these sets, plus the overlap

with respect to F of the peers that belong to the intersection of the sets SP1, … , SPn-1 and the peers that belong to set SPn but not to the intersection of SP1, … , SPn. From this sum we should subtract the overlap with respect to F of those peers that belong to SP1∩…∩ SPn, to SPn - SP1∩…∩ SPn and to SP1∩… ∩SPn-1, since this overlap has already been counted twice.

3.8 OVERLAP AND CARDINALITY ESTIMATION OF QUERY PLANS

During the query routing and planning phase, the peers that can contribute to the answer of a specific query fragment F are contacted and query plans that can answer F are created. Apart from the case where one peer can answer the whole query and as a result the corresponding query plan will involve no joins between fragments, in most cases, query plans involve one or more joins between fragments exported by different peers. In this section we will present formulae for the overlap and cardinality estimation of query plans.

3.8.1 OVERLAP ESTIMATION

Let us assume that we have a query fragment F_{1236} = r3 ⋈ r1 ⋈ r2 ⋈ r6, where fragments F_{13}=r3 ⋈ r1 and F_{26}=r2 ⋈ r6 are joined on class C2. Thus, we consider as reference set for the overlap computation the total cardinality of class C2 in the PDMS, i.e.||C2||.

Let us assume that there is another peer P8 which exports fragment F_{13}=r3 ⋈ r1. Then, according to Figure 4, two query plans that can answer to this fragment are QP1 = F_{13}@P9 ⋈ F_{26}@P15 and QP2 = F_{13}@P8 ⋈ F_{26}@P14. Then, their overlap will be computed similarly to the cardinality estimation for joins of fragments that belong to different peers presented in section 3.9.1.

Overlap (F_{13}@P9 ⋈ F_{26}@P15, F_{13}@P8 ⋈ F_{26}@P14) = $Overlap_{P8,P9}(F_{13})$ * $Overlap_{P14,P15}(F_{26})$ / ||C2|| ***(3.27)***

It is clear that formula (3.27) also holds for arbitrary complex fragments. If peers Pi and Pi' export fragment F1 and peers Pj and Pj' export fragment F2, while F1 and F2 are joined on class C, then

Overlap (F1@Pi ⋈ F2@Pj, F1@Pi' ⋈ F2@Pj') = $Overlap_{Pi,Pi'}(F1)$ * $Overlap_{Pj,Pj'}(F2)$ / ||C|| ***(3.28)***

To generalize, if we assume that peers P1 and P1' export fragment F1, peers P2 and P2' export fragment F2, … , peers Pn and Pn' export fragment Fn, then the overlap between the query plan of peers P1, …, Pn and that of P1', …, Pn', will be

$$Overlap(F1@P1\times\ldots\times Fn@Pn, F1@P1'\times\ldots\times Fn@Pn') = \prod_i Overlap_{Pi,Pi'}(Fi) / \prod_j ||C_j|| \quad \textbf{(3.29)}$$

where i = 1, …, n and j = 1, …, n-1.

Similarly, if we consider arbitrary sets of peers SP1, SP1' that export fragment F1, etc, formula (3.29) will become

$$Overlap(F1@SP1\times\ldots\times Fn@SPn, F1@SP1'\times\ldots\times Fn@SPn') = \prod_i Overlap_{SPi,SPi'}(Fi) / \prod_j ||C_j|| \quad \textbf{(3.30)}$$

where i = 1, …, n and j = 1, …, n-1.

In general for arbitrary sets of peers SP11, …, SPn1, …, SP1m, …, SPnm we have

$$Overlap(F1@SP1,1\times\ldots\times Fn@SPn,1, \ldots, F1@SP1,m\times\ldots\times Fn@SPn,m) = \prod_i Overlap_{SPi,1,\ldots,SPi,m}(Fi) / \prod_j ||C_j|| \quad \textbf{(3.31)}$$

where i = 1, …, n and j = 1, …, n-1.

3.8.2 CARDINALITY ESTIMATION

Let us consider again fragments $F1_{P9}=r3_{P9} \bowtie r1_{P9}$ and $F2_{P15}=r2_{P15} \bowtie r6_{P15}$. These fragments join on their common class C2. In the independence overlap case, the overlap of P9 and P15 with respect to C2 is the probability that a random C2 instance of the PDMS belongs to both peers. The various overlap cases can be are similar to Table 3.2.

We can easily understand that the case of two peers whose fragments join on subsumed classes is similar. The overlap in this case is computed as in Table 3.3. Moreover, it is clear that the same formulae for overlap estimation also hold when the fragments of the peers that join on a common class are more complex than a single property. In general, if peer Pi exports fragment F1, peer Pj exports fragment F2 and these fragments join on their common class C, then the overlap cardinality between these peers using formula (3.3) is

$$Overlap_{Pi,Pj}(C) = overlap_{Pi,Pj}(C) * ||C||$$

Then, the cardinality of the join between F1 and F2 can be given by the following formula :

$$|F1_{P9} \bowtie F2_{P15}| = |F1|_{P9} * |F2|_{P15} * Overlap_{P9,P15}(C2) / (|C2|_{P9} * |C2|_{P15}) \quad \textbf{(3.32)}$$

where $Overlap_{P9,P15}(C2)$ is computed by the formula

$$Overlap_{P9,P15}(C2) = overlap_{P9,P15}(C2) * ||C2|| = |C2|_{P9} * |C2|_{P15} / ||C2||$$

and thus, formula (3.27) can be written equivalently as

$|F1_{P9} \bowtie F2_{P15}| = |F1|_{P9} * |F2|_{P15} / ||C2||$ ***(3.33)***

Formulae (3.32) and (3.33) also hold for arbitrary, more complex fragments. Let us assume that peer Pi exports fragment Fi and peer Pj exports fragment Fj which join on class C2. Then, formulae (3.32) and (3.33) will become

$|Fi_{Pi} \bowtie Fj_{Pj}| = |Fi|_{Pi} * |Fj|_{Pj} * Overlap_{Pi,Pj}(C2) / (|C2|_{Pi} * |C2|_{Pj})$ =>

$|Fi_{Pi} \bowtie Fj_{Pj}| = |Fi|_{Pi} * |Fj|_{Pj} / ||C2||$

These formulae can be generalized for arbitrary sets of peers. Let us consider set SP1 of peers that export fragment Fi and set SP2 of peers that export fragment Fj. Then

$|Fi_{SP1} \bowtie Fj_{SP2}| = |Fi|_{SP1} * |Fj|_{SP2} / ||C2||$ ***(3.34)***

If we have an arbitrary number of subfragments F1, F2, … , Fn over arbitrary disjoint sets of peers SP1, SP2, … , SPn joined over a set of classes C1, C2, … , Cn-1, the cardinality of the join of all subfragments will be

$|F1_{SP1} \bowtie F2_{SP2} \ldots \bowtie Fn_{SPn}| = \prod_i |Fi|_{SPi} / \prod_j ||C_j||$ ***(3.35)***

where i = 1, … , n and j = 1, … , n-1.

3.8.3 UNIONS OF JOINS

The query plans we considered in sections 3.8.1. and 3.8.2. are equivalent to a union of joins of the same relation fragments, exported by different peers. For example, let us take the PDMS fragment F of Figure 5, consisting of properties r1, r3, r2 and r6. A fragmentation of F involving one join, is $F_{13} \bowtie F_{26}$, where $F_{13} = r1 \bowtie r3$ and $F_{26} = r2 \bowtie r6$. Assuming that peer P8 exports fragment F_{13} and taking Figure 3 into account we can see that two peers that can answer fragment F_{13} are P8 and P9, while peers P14 and P15 can answer fragment F_{26}. So, the query plans that can answer fragment F_{1236} form a union of joins as follows:

$$F_{13\{P8,P9\}} \bowtie F_{26\{P14,P15\}} = F1_{P8} \bowtie F3_{P8} \bowtie F2_{P14} \bowtie F6_{P14} \cup$$
$$F1_{P8} \bowtie F3_{P8} \bowtie F2_{P15} \bowtie F6_{P15} \cup$$
$$F1_{P9} \bowtie F3_{P9} \bowtie F2_{P14} \bowtie F6_{P14} \cup$$
$$F1_{P9} \bowtie F3_{P9} \bowtie F2_{P15} \bowtie F6_{P15}$$

As a consequence, the union of any number of such plans has the same form. Thus, the number of new results produced by a plan QP after a set of already executed plans {QPi}, where i = 1, …, n, is:

$|NewResults(QP,\{QPi\})| = |QP \cup QP1 \cup \ldots \cup QPn| - |QP1 \cup \ldots \cup QPn|$ ***(3.36)***

As far as the cardinality of such a union of joins is concerned, we can compute it similarly to formula (3.22) for the cardinality of a fragment F in n peers :

$|\cup i\ F1_{SP1,i} \times \ldots \times Fn_{SPn,i}| = \sum i\ |F1_{SP1,i} \bowtie \ldots \bowtie Fn_{SPn,i}| - \sum ij\ Overlap(F1_{SP1,i} \bowtie \ldots \bowtie Fn_{SPn,i}, F1_{SP1,j} \bowtie \ldots \bowtie Fn_{SPn,j}) + \ldots + (-1)^{m-1} Overlap(F1_{SP1,1} \bowtie \ldots \bowtie Fn_{SPn,1}, \ldots, F1_{SP1,m} \bowtie \ldots \bowtie Fn_{SPn,m})$ ***(3.37)***

3.8.4 OVERLAP ESTIMATION EXAMPLE

In the previous sections we presented the overlap computations in the case of disjoint and non – disjoints sets of peers. We will provide as an example the overlap estimation of the plans of Figure 4. Taking Table 2.3 into account, the plans of Figure 4 are the following :

$QP1 = F_{1236}$

$QP2 = F_{123\{P14\}} \bowtie F_{6\{P14,P15\}}$

$QP3 = F_{126\{P14\}} \bowtie F_{3\{P9,P14\}}$

$QP4 = F_{13\{P9,P14\}} \bowtie F_{26\{P14,P15\}}$

$QP5 = F_{12\{P12,P13,P14\}} \bowtie F_{3\{P9,P14\}} \bowtie F_{6\{P14,P15\}}$

$QP6 = F_{26\{P14,P15\}} \bowtie F_{1\{P9,P12,P13,P14\}} \bowtie F_{3\{P9,P14\}}$

$QP7 = F_{13\{P9,P14\}} \bowtie F_{2\{P12,P13,P14,P15\}} \bowtie F_{6\{P14,P15\}}$

$QP8 = F_{1\{P9,P12,P13,P14\}} \bowtie F_{2\{P12,P13,P14,P15\}} \bowtie F_{3\{P9,P14\}} \bowtie F_{6\{P14,P15\}}$

As we can see in Figure 3, the peers that we consider are not disjoint with each other. In favor of simplicity we will show how we can compute the overlap for some pairs of plans of Figure 4, considering also the cardinality information of Table 3.1. :

$Overlap(QP1, QP2) = |F_{1236\{P14\}}| = 3$

$Overlap(QP1, QP3) = |F_{1236\{P14\}}| = 3$

$Overlap(QP1, QP4) = |F_{1236\{P14\}}| = 3$

$Overlap(QP1, QP5) = |F_{1236\{P14\}}| = 3$

$Overlap(QP1, QP6) = |F_{1236\{P14\}}| = 3$

$Overlap(QP1, QP7) = |F_{1236\{P14\}}| = 3$

$Overlap(QP1, QP8) = |F_{1236\{P14\}}| = 3$

As we can see in Table 3.1, the total distinct number of F_{1236} instances in the PDMS is 13. Thus, we can express the overlap between plans QP1 – QP8 as probability as follows :

overlap(QP1, QP2) = 3 / 13 = 0,23

overlap(QP1, QP3) = 3 / 13 = 0,23

overlap(QP1, QP4) = 3 / 13 = 0,23

overlap(QP1, QP5) = 3 / 13 = 0,23

overlap(QP1, QP6) = 3 / 13 = 0,23

overlap(QP1, QP7) = 3 / 13 = 0,23

overlap(QP1, QP8) = 3 / 13 = 0,23

To compute the overlap between more pairs of plans and among three of four plans, more complex formulae need to be developed.

3.9 ACCURACY OF OVERLAP ESTIMATIONS

In all the previous sections, the symbol ||C|| represents the number of distinct instances in the system. However, it is difficult to maintain an accurate estimation for the global number of disjoint class instances in the PDMS. To properly estimate ||C|| we should eliminate the instance overlaps between peers. The problem is that estimating these overlaps is difficult.

A possible realistic approximation of ||C|| is given by the following identity:

$$| F_{12}@P_1| = |F_1@P_1 \times F_2@P_1| = |F_1|_{P1} * |F_2|_{P1} / ||C|| \Rightarrow ||C|| = |F_1|_{P1} * |F_2|_{P1} / | F_{12}@P_1|$$

In the above formula, F_{12} is a fragment composed of sub-fragments F_1 and F_2 joined on C, so we can apply formula (3.19) for the evaluation of the sub-fragments on the same peer P_1. Since cardinalities of fragments on a peer are known, one may deduce the following estimation of |C|:

$$||C|| = |F_1|_{P1} * |F_2|_{P1} / | F_{12}|_{P1} \qquad \textbf{\textit{(3.38)}}$$

By applying formula (3.38) for various fragments and peers, the average of these values may be considered as a good global estimation of ||C||.

3.10 RELATED WORK

In this section we compare our formulae for overlap estimation with respect to related work proposed in the literature.

A mediator – based information system is presented in **[4]**, where a universal relation model is used as a global schema for formulating user queries against relational data sources. This global schema is essentially the union of all attributes of source relations. On the contrary, we

consider a PDMS where peers do not export relational attributes, but arbitrary complex RDF/S schema fragments. In **[4]**, the overlap cases considered are the same as ours:

- ✓ Two sources are disjoint if they do not provide any common tuples of the universal relation.
- ✓ Two sources are independent if there is no (known) dependency between the tuples of the universal relation for which the sources provide data. i.e. there is some coincidental overlap between the sources.
- ✓ Two sources have quantified overlap, when the exact degree of overlap, i.e. the number of their common universal relation tuples is known.
- ✓ One source is contained in another if every tuple of the universal relation the one source provides is also provided by the other source. However, one source may provide different attributes in a tuple than the other.

The major underlying assumptions made in **[4]** are : a) the probability that a source has a null value for an attribute of a certain tuple is independent of the probability that another source has a null value for the same attribute of the tuple representing the same real world entity, b) there is uniform distribution of attribute values for all attributes. In our work, we rely on optional RDF properties bypassing the need for handling null values while most of the interesting joins in this setting are over the domain of resource URIs (i.e., strings) for which no uniform distribution assumption is made.

Estimating overlap in our model is a more tricky task than in **[4]**, since we need to calculate the overlap of two or more peers with respect to a specific PDMS schema fragment (i.e. a single class, a property or arbitrary property joins) while in **[4]** overlap refers only to the number of common universal relation tuples between sources. As in **[4]**, when more than two peers exporting the same fragment, we make the realistic assumption of independence overlap in the employed data quality metrics unless additional knowledge is provided by the peers.

The overlaps (i.e., probabilities) computed by our formulae are likely to result in smaller estimations than the ones provided by **[4]**. This is due to the fact that in our setting we do not just have to compute the number of common instances among sources, but we should also take into account the probability that common class instances in the peers are also related through RDF properties, and then these property instances could be joined to form more complex fragment

instances, etc. Thus, overlap formulae are formed as a product of probabilities and as a result, they are likely to give smaller estimations.

The mediator – based information system presented in **[5]**, aims to first access the sources that have a higher probability of containing an answer to a user query. For this reason, they estimate the overlap of collections in the mediated schema. Each collection can be viewed as a unary relation whose extension is a set of objects. Properties of objects are modeled by a set of attributes, either single or multi – valued. The extension of a collection is not necessarily the union of all the objects found in a given set of relevant data sources. For example, a collection named Databases may denote the set of published papers in the field of databases, and does not depend on the specific sources available to the system at a given time.

The authors take three cases of overlap between collections: a) one collection is a superset of another, b) two collections are disjoint, c) there is some overlap between a pair of collections. Conditional probabilities are used to specify overlap between collections in the mediated schema. A conditional probability of an incidence A, given incidence B is the probability that A may happen, given that B has happened and is denoted as $P(A|B) = \frac{P(A \wedge B)}{P(B)}$. For example, the overlap between collections c1 and c2, denoted as P(c1|c2), is the conditional probability that an object belonging to c2 also belongs to c1. Every source S supported by the mediator is described by a schema query Qs.

In order to compute overlap between information sources, only the case of independence between them is taken into account. Thus, the overlap of sources S1 and S2 (described by schema queries Q1 and Q2 respectively) with respect to a query Q is calculated as the product of the conditional probabilities P(S1|Q1) * P(S2|Q2) * P(Q1^Q2|Q), where the factor P(Q1^Q2|Q) is the conditional probability that an object belonging to both schema queries Q1 and Q2 also belongs to query Q. However, no information is provided on how this probabilistic information can be obtained. In our framework, overlap between two or more peers is also computed using conditional probabilities. Let us consider the case of peers P1 and P5 of Figure 2. They both export property r1 with domain class C1 and range class C2. The independence overlap is the product of the conditional probabilities that the C1 and C2 instances of each peer, also form r1 instances.

In **[23]**, the notion of overlap arises in the computation of plan coverage. The model and assumptions used for overlap estimation between sources in this work, are the ones described in **[5]**, and no further reference is made throughout the article.

The goal of the meta – search engine of Web sources presented in **[29]** is to maximize the coverage of mediated queries, while keeping the mediator cost under check. The relational data model is considered for the contents of the sources. In this framework, qualitative, rather than quantitative overlap information is used, since the latter is usually hard to obtain and may be inaccurate. The overlap cases considered in **[29]** are quite similar to ours. The equivalence overlap case they consider is a specialization of the quantified overlap case we have defined, when overlap between the peers is 100%. For more than two peers, independence is assumed for the overlap estimation, like in our work.

Chapter 4

PEER DATA QUALITY METRICS

To reduce the very large planning space we should be able to rank the peers contributing to a plan, and thus the plans themselves, according to data quality metrics allowing to discard plans producing poor quality query results (according to a threshold either set by the user or the system). To this end we consider data quality metrics such as *coverage*, *density* and *completeness* of the view instances published by the peers with respect to the PDMS schema and its virtual instantiation. In the following, we give the definitions of the quality metrics, we propose formulae for their calculation in different cases and we provide some examples.

4.1 BASIC DEFINITIONS

Before we define the quality metrics we should point out that we divide them in two categories: global and local. A global quality metric of a peer takes into account its characteristics (e.g. cardinality) with respect to the same characteristics for all peers in the PDMS. Such a metric is coverage. On the other hand, a local quality metric refers to a peer's characteristic with reference to the peer itself, or to a set of two or more peers, but not to the whole PDMS. Such a metric is density. Completeness is a global quality metric that combines both coverage and density. The quality metrics that will be defined and described are used in the context of a single peer as well as in the context of two or more peers.

The reference set with respect to which we will define our quality metrics for an arbitrary fragment F is the maximum number of F instances that can be retrieved in our PDMS. This is expressed by the product of all the instances of the classes involved in a fragment.

In the following, we denote as $|F|_P$ the cardinality of F in P, as $|F|_{Pmax}$ the Cartesian product of the class instances that form instances of fragment F in peer P and as ||F|| the number of all the F instances in the PDMS, which is the maximum number of F instances that can be retrieved if all the instances of the classes that F comprises were related to each other.

Definition 4.1 *: The coverage of a peer database P with respect to a specific PDMS fragment F is the ratio of the maximum number of fragment F instances one expects to obtain through its view (i.e., when all class instances are related through the properties declared by a peer schema*

fragment) to the maximum total number of F instances published in the PDMS. The coverage of a peer database with respect to a PDMS schema fragment F is a global quality metric, since it takes into account the maximum total cardinality of this fragment in the PDMS.

The coverage formula of peer P with respect to fragment F, which comprises classes Cj, j = 1, …, n is

$$\text{cov}_P(F) = \frac{|F|_{P\max}}{||F||} = \prod_j |C_j|_P / \prod_j ||C_j|| \quad \textbf{(4.1)}$$

In other words, we could say that the coverage of a peer P with respect to fragment F is a global estimation of the maximum percentage of F instances we could obtain from peer P if all the class instances that form fragment F were connected through schema properties. We should state that high coverage implies an increased interest in considering peer P in a plan computing F instances.

Definition 4.2*: The density of a peer database P with respect to a specific PDMS fragment F is the ratio of the number of fragment F instances exported by a peer (i.e., when materializing its view at a certain point in time) to the maximum number of fragment F instances in this peer, which is the product of all the class instances that form F instances in this peer. Density is a local quality metric, taking into account the cardinality of this fragment instances exported by a specific peer.*

The density formula of peer P with respect to fragment F, which comprises classes Cj, j = 1, …, n is

$$\text{den}_P(F) = \frac{|F|_P}{|F|_{P\max}} = |F|_P / \prod_j |C_j|_P \quad \textbf{(4.2)}$$

In other words we could say that density of a peer P with respect to fragment F is a local estimation of the actual percentage of F instances we could obtain from peer P with respect to the maximum number of expected F instances, i.e. the percentage of the instances of the classes which F comprises that are related to each other to form instances of fragment F. We should state that high density implies an increased interest in considering peer P in a plan computing also sub F instances (at least of the same completeness).

Definition 4.3*: The completeness of a peer P with respect to a specific PDMS fragment F captures the ratio of the cardinality of this fragment in peer P to the maximum total cardinality of this fragment in the PDMS. Completeness of a peer database is a global quality metric. When the fragment is a class, we do not distinguish between coverage and density and we only define the notion of completeness.*

The completeness formula of peer P with respect to fragment F is

$$com_P(F) = \frac{|F|_P}{||F||} = |F|_P / \prod_j ||C_j|| \qquad \textbf{(4.3)}$$

In other words, the completeness of a peer P with respect to a fragment F expresses the probability that a random instance of F in the PDMS belongs to this peer base. Since we have already defined the notions of coverage, density and completeness, in each of the following sections we will express overlap with respect to these metrics. In addition, we will present and explain formulae for the estimation of the quality metrics addressed above in the case of two or more peers. We should state that completeness is a metric than combines the estimation about the maximum number of fragment instances that a peer can return (coverage) with the estimation about the actual fragment instances contained in the peer base (density). Besides, a global completeness metric does not distinguish between combinations of high or low coverage and density.

In addition, the following theorem holds for each overlap case of chapter 4:

Theorem 4.1*: When a peer P exports a fragment F of the PDMS schema (where F can be either a simple or a complex fragment) then the completeness of P with respect to F, denoted as $com_P(F)$ is computed as the product of the coverage of P with respect to F (denoted as $cov_P(F)$) and the density of P with respect to F (denoted as $den_P(F)$).*

Proof *: Formula (4.3) can be written as*

$$com_P(F) = \frac{|F|_P}{||F||} = \frac{|F|_{P\max}}{||F||} * \frac{|F|_P}{|F|_{P\max}} = \prod_j |C_j|_P / \prod_j ||C_j|| * |F|_P / \prod_j |C_j|_P = cov_P(F) * den_P(F)$$

if F involves joins over a set of classes Cj, where j = 1, ,… n.

In the following sections we will show that Theorem 4.1 is applied in any case of a fragment for which coverage and density can be defined.

4.2 DATA QUALITY OF FRAGMENTS IN ONE PEER

In section 4.1 we defined coverage, density and completeness of a schema fragment that is exported by a peer. In this section we present formulae for these quality metrics when the peer fragment is a single class, a single property or a join of two or more properties.

4.2.1 QUALITY METRICS FOR A SINGLE CLASS

As shown in Figure 3, peer P9 exports class C1. As mentioned in Definition 4.3, in the case of a single class completeness is the only metric used. The reference set for completeness computations is the total number of C1 instances in the PDMS, i.e. ||C1||. As a result, from Definition 4.3 we have :

$$com_{P9}(C1) = \frac{|C1|_{P9}}{||C1||} \quad \textbf{(4.4)}$$

4.2.2 QUALITY METRICS FOR A SINGLE PROPERTY

In Figure 7 we can see that peer P9 exports property r1 with domain class C1 and range class C2. The reference set for these quality metrics is the product of all C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2|| . This is the maximum number of r1 instances that can be retrieved in the PDMS. Using definitions 4.1, 4.2 and 4.3 we have :

$$cov_{P9}(r1) = \frac{|C1|_{P9} * |C2|_{P9}}{||C1|| * ||C2||} \quad \textbf{(4.5)}$$

$$den_{P9}(r1) = \frac{|r1|_{P9}}{|C1|_{P9} * |C2|_{P9}} \quad \textbf{(4.6)}$$

$$com_{P9}(r1) = \frac{|r1|_{P9}}{||C1|| * ||C2||} \quad \textbf{(4.7)}$$

4.2.3 QUALITY METRICS FOR COMPLEX FRAGMENTS

In Figure 3 peer P12 exports fragment F1, which consists of properties r1 and r2 joined on their common class C2. Property r1 has domain class C1 and range class C2, while property r2 has domain class C2 and range class C3. The reference set used is the product of all the instances of the classes that fragment F1 comprises, i.e. ||C1|| * ||C2|| * ||C3||. This is the maximum number of F1 instances that can be retrieved in the PDMS. Then, the quality metrics of F1 are given by the following formulae:

$$cov_{P12}(F1) = \frac{|C1|_{P12} * |C2|_{P12} * |C3|_{P12}}{||C1|| * ||C2|| * ||C3||} \qquad \textbf{(4.8)}$$

$$den_{P12}(F1) = \frac{|F1|_{P12}}{|C1|_{P12} * |C2|_{P12} * |C3|_{P12}} \qquad \textbf{(4.9)}$$

$$com_{P12}(F1) = \frac{|F1|_{P12}}{||C1|| * ||C2|| * ||C3||} \qquad \textbf{(4.10)}$$

The formulae (4.8), (4.9) and (4.10) can be used for more complex peer fragments as well. For example, in Figure 3 peer P14 exports fragment F2, which consists of the join of properties r1, r2 and r3 and classes C1, C2, C3, C4. The coverage, density and completeness formulae of F2 are shown below :

$$cov_{P14}(F2) = \frac{|C1|_{P14} * |C2|_{P14} * |C3|_{P14} * |C4|_{P14}}{||C1|| * ||C2|| * ||C3|| * ||C4||}$$

$$den_{P14}(F2) = \frac{|F2|_{P14}}{|C1|_{P14} * |C2|_{P14} * |C3|_{P14} * |C4|_{P14}}$$

$$com_{P14}(F2) = \frac{|F2|_{P14}}{||C1|| * ||C2|| * ||C3|| * ||C4||}$$

In general, we can say that when a peer P exports a fragment F consisting of k properties joined together, relating classes C1, C2, …, Ck+1, the quality metrics formulae for this fragment using formulae (4.8), (4.9) and (4.10) are given below. The reference set used is the product of all the instances of the classes that fragment F comprises, i.e. ||C1|| * ||C2|| * … *||C_{k+1}||. The

maximum number of F instances in peer P is considered to be the product of the class instances that F comprises, i.e. $|C1|_P * |C2|_P * \ldots * |C_{k+1}|_P$

$$cov_P(F) = \frac{|C1|_P * |C2|_P * ... * |Ck+1|_P}{||C1|| * ... * ||C_{k+1}||} \quad \textbf{(4.11)}$$

$$den_P(F) = \frac{|F|_P}{|C1|_P * |C2|_P * ... * |Ck+1|_P} \quad \textbf{(4.12)}$$

$$com_P(F) = \frac{|F|_P}{||C1|| * \ldots * |C_{k+1}||} \quad \textbf{(4.13)}$$

4.3 UNION OF PEER FRAGMENT INSTANCES

4.3.1 PEERS PUBLISHING THE SAME CLASS

In Figure 3, we can see that peers P9 and P12 export instances of C1. The reference set for completeness computations is the total number of C1 instances in the PDMS, i.e. ||C1||. As stated previously in Definition 4.3, when two or more peers export the same class only the completeness quality metric is used. So, the completeness formula of the union of C1 in P9 and P12 is

$$com_{P9,P12}(C1) = \frac{|C1|_{P9} + |C1|_{P12} - Overlap_{P9,P12}(C1)}{||C1||} \quad \textbf{(4.14)} \quad =>$$

$$com_{P9,P12}(C1) = com_{P9}(C1) + com_{P12}(C1) - overlap_{P9,P12}(C1) \quad \textbf{(4.15)}$$

The idea behind the completeness formula for the union is simple: since both P9 and P12 export the same class C1, the completeness of the union will be the ratio of the sum of the distinct C1 instances in the two peers (this is why we subtract the C1 instances that are counted twice, as they appear in both peers, i.e. the overlap of C1 in the two peers) to the total number of C1 instances in the PDMS, i.e. ||C1||. We should point out that in formula ***(4.14)*** the overlap is expressed as cardinality, while in formula ***(4.15)*** it is expressed as a probability.

Table 4.1 shows the overlap cases presented in Table 3.1 with respect to the completeness formula of a class ***(4.4)*** :

Overlap Cases	Overlap Estimation
Disjointness	$overlap_{P9,P12}(C1) = 0$
Independence	$overlap_{P9,P12}(C1) = com_{P9}(C1) * com_{P12}(C1)$
Quantified Overlap	$overlap_{P9,P12}(C1) = X$ where X is a known probability
Containment (e.g. of $C1_{P9}$ in $C1_{P12}$)	$overlap_{P12,P9}(C1) = \frac{com_{P9}(C1)}{com_{P12}(C1)}$ $overlap_{P9,P12}(C1) = 1$

Table 4.1 : Overlap cases for two peers exporting the same class

Table 4.2 presents the completeness of the union of two peers with respect to a class that both export. In fact, it presents formula (4.14) for the different overlap cases of Table 4.1 :

Overlap Cases	Completeness Estimation
Disjointness	$com_{P9,P12}(C1) = com_{P9}(C1) + com_{P12}(C1)$
Independence	$com_{P9,P12}(C1) = com_{P9}(C1) + com_{P12}(C1) - com_{P9}(C1) * com_{P12}(C1)$
Quantified Overlap	$com_{P9,P12}(C1) = com_{P9}(C1) + com_{P12}(C1) - X$ where X is a known probability
Containment (e.g. of $C1_{P9}$ in $C1_{P12}$)	$com_{P9,P12}(C1) = com_{P9}(C1) + com_{P12}(C1) - 1$ $com_{P12,P9}(C1) = com_{P9}(C1) + com_{P12}(C1) - \frac{com_{P9}(C1)}{com_{P12}(C1)}$

Table 4.2 : Completeness for two peers exporting the same class

The formulae shown in Table 4.2 can be easily explained. When there is no overlap of C1 in peers P9 and P12, the completeness of C1 is the sum of completeness of C1 in each peer. In the cases of independence and quantified overlap, we should add the completeness of C1 in the two peers and subtract the probability of their common C1 instances. As explained earlier, in the sum of completeness probabilities of C1 in the two peers, we have counted twice the C1 instances that are common in them. We need to correct this by subtracting the probability of the common C1 instances between P9 and P12, in order to calculate only once those C1 instances which appear in both peers. In the case of containment overlap, we have two types for completeness, one for each overlap case we consider. It is clear that in all cases in which there is some overlap of C1 in the two peers, the value of completeness becomes smaller as the value of overlap gets greater.

4.3.2 PEERS PUBLISHING SUBSUMED CLASSES

As shown in Figure 3, peer P9 exports instances of class C1, while P13 exports instances of C7, which is a subclass of C1. The reference set for completeness computations is the total number of C1 instances in the PDMS, i.e. ||C1||, since all the instances of class C7 can also be exported as C1 instances, and as a result all the instances of C7 are involved in ||C1||. As in the previously, only the completeness quality metric is used. So, similarly to formula ***(4.14)***, we have

$$com_{P9,P13}(C1UC7) = \frac{|C1|_{P9} + |C7|_{P13} - Overlap_{P9,P13}(C1UC7)}{||C1||} \quad \textbf{(4.16)} \Rightarrow$$

$$com_{P9,P13}(C1UC7) = \frac{|C1|_{P9}}{||C1||} + \frac{|C7|_{P13}}{||C1||} - overlap_{P9,P13}(C1UC7) \quad \textbf{(4.17)}$$

The idea behind the completeness formula for the union is simple: Since C7 is a subclass of C1, the C7 instances of P13 are also used to answer queries concerning C1. So, the completeness of the union will be the ratio of the sum of the distinct number of C1 and C7 instances in the two peers (this is why we subtract the C5 instances that are counted twice, as they appear in both peers, i.e. the overlap of C7 in the two peers) to the total number of C1 instances in the PDMS, i.e. ||C1||. The completeness formula (4.17) can also be written as

$com_{P9,P13}(C1UC7) = com_{P9}(C1) + com_{P13}(C7) - overlap_{P9,P13}(C1UC7)$ ***(4.18)***

Table 4.3 shows the overlap cases presented in Table 3.2 with respect to the class completeness formula (4.4) :

Overlap Cases	**Overlap Estimation**
Disjointness	$overlap_{P9,P13}(C1UC7) = 0$
Independence	$overlap_{P9,P13}(C1UC7) = com_{P9}(C1) * com_{P13}(C7)$
Quantified Overlap	$overlap_{P9,P13}(C1UC7) = X$ where X is a known probability
Containment (e.g. of $C7_{P13}$ in $C1_{P9}$)	$overlap_{P9,P13}(C1UC7) = \frac{com_{P13}(C7)}{com_{P9}(C1)}$ $overlap_{P2,P1}(C1UC5) = 1$

Table 4.3 : Overlap cases for two peers exporting subsumed classes

Table 4.4 presents formula (4.16) in peers P9 and P13, for all the overlap cases of Table 4.3 :

Overlap Cases	**Completeness Estimation**
Disjointness	$com_{P9,P13}(C1UC7) = com_{P9}(C1) + com_{P13}(C7)$
Independence	$com_{P9,P13}(C1UC7) = com_{P9}(C1) + com_{P13}(C7) - com_{P9}(C1) * com_{P13}(C7)$
Quantified Overlap	$com_{P9,P13}(C1UC7) = com_{P9}(C1) + com_{P13}(C7) - X$ where X is a known probability
Containment (e.g. of $C7_{P13}$ in $C1_{P9}$)	$com_{P13,P9}(C1UC7) = com_{P9}(C1) + com_{P13}(C7) - 1$ $com_{P9,P13}(C1UC7) = com_{P9}(C1) + com_{P13}(C7) - \frac{com_{P13}(C7)}{com_{P9}(C1)}$

Table 4.4 : Completeness for two peers exporting subsumed classes

The formulae shown in Table 4.4 can be easily explained. When there is no overlap of C1 and C7 in peers P9 and P13, the completeness of C1 is the sum of completeness of C1 in peer P9 and C7 in P13. In the case of independence overlap, we should add the completeness of C1 and C7 in the two peers and subtract the probability of their common instances. As explained earlier, in the sum of completeness probabilities of C1 in P9 and C7 in P13, we have counted twice the C7 instances that are common in them. We need to correct this by subtracting the probability of the common C1 and C7 instances between P9 and P13, in order to calculate only once those C7 instances that appear in both peers. In the case of containment overlap, if the C7 instances of peer P13 are completely contained in the C1 instances of peer P9, we have two formulae for completeness, depending on which peer we consider completeness for. It is clear that in all cases in which there is some overlap of C1 and C7 in the two peers, the value of completeness becomes smaller as the value of overlap gets greater.

4.3.3 PEERS PUBLISHING THE SAME PROPERTY

In the case of classes, there is only the completeness metric that we consider. However, when it comes to properties and more complex fragments, we have the notions of coverage and density. Until now we only considered the coverage and density metrics for one peer. In the following sections we will present formulae for the estimation of coverage, density and completeness of

properties or more complex fragments in two or more peers. In all cases, we will show that the completeness formula will be the product of the coverage and density formula.

4.3.3.1 PROPERTIES WITH THE SAME DOMAIN/RANGE

As shown previously in Figure 3, peers P9 and P12 both export property r1 with domain class C1 and range class C2. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2||. This is the maximum number of r1 instances that can be retrieved in the PDMS. The coverage formula of the union of r1 in P9 and P12 is

$$cov_{P9,P12}(r1) = \frac{|C1|_{P9} * |C2|_{P9} + |C1|_{P12} * |C2|_{P12} - Overlap_{P9,P12}(C1 \times C2)}{||C1|| * ||C2||} \quad \textbf{(4.19)} \Rightarrow$$

$$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1) - overlap_{P9,P12}(C1) * overlap_{P9,P12}(C2) \quad \textbf{(4.20)}$$

The idea behind the coverage formula for the union is simple: according to definition 4.1 the coverage of property r1 in peers P9 and P12 is the ratio of the maximum possible number of distinct r1 instances in the two peers (i.e. the sum of the product of the C1 and C2 instances in the two peers if we subtract their overlap of C1 and C2) to the maximum possible number of r1 instances in the PDMS (i.e. the product of the total C1 and C2 instances in the PDMS).

The formula for density of r1 in peers P9 and P12 will be

$$den_{P9,P12}(r1) = \frac{|r1|_{P9} + |r1|_{P12} - Overlap_{P9,P12}(r1)}{|C1|_{P9} * |C2|_{P9} + |C1|_{P12} * |C2|_{P12} - Overlap_{P9,P12}(C1 \times C2)} \quad \textbf{(4.21)}$$

Formula (4.21) stems from definition 4.2 : density of peers P9 and P12 with respect to property r1 is the ratio of the distinct number of r1 instances in the two peers (i.e. the sum of their r1 instances if we subtract their overlap with respect to r1) to the maximum possible number of distinct r1 instances in P9 and P12.

As far as completeness of r1 in peers P9 and P12 is concerned, we have:

$$com_{P9,P12}(r1) = cov_{P9,P12}(r1) * den_{P9,P12}(r1) = \frac{|r1|_{P9} + |r1|_{P12} - Overlap_{P9,P12}(r1)}{||C1|| * ||C2||} \quad \textbf{(4.22)} \Rightarrow$$

$$com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1) - overlap_{P9,P12}(r1) \quad \textbf{(4.23)}$$

As we can see, the completeness formula is formed as the product of coverage and density and as stated by definition 4.3 it is the ratio of the number of distinct r1 instances in peers P9 and P12 to the total maximum number of r1 instances in the PDMS. Table 4.5 shows the overlap cases presented in Table 3.4 with respect to the completeness metrics formulae (4.4) and (4.7):

Overlap of C1 and C2 in peers P9, P12	**Overlap Estimation Formulae for C1, C2**	**Overlap Estimation Formulae for property r1**
Disjointness of C1, C2	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = 0$	$overlap_{P9,P12}(r1) = 0$
Disjointness of C1 and independence of C2	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = com_{P9}(C2) * com_{P12}(C2)$	$overlap_{P9,P12}(r1) = 0$
Disjointness of C1 and quantified overlap of C2	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = X$, where X is a known probability	$overlap_{P9,P12}(r1) = 0$
Disjointness of C1 and containment of $C2_{P8}$ in $C2_{P1}$	$overlap_{P9,P12}(C1) = 0$ $overlap_{P9,P12}(C2) = \frac{com_{P12}(C2)}{com_{P9}(C2)}$ $overlap_{P12,P9}(C2) = 1$	$overlap_{P9,P12}(r1) = 0$
Independence of C1, C2	$overlap_{P9,P12}(C1) = com_{P9}(C1) * com_{P12}(C1)$ $overlap_{P9,P12}(C2) = com_{P9}(C2) * com_{P12}(C2)$	$overlap_{P9,P12}(r1) = com_{P9}(r1) * com_{P12}(r1)$
Containment of $C1_{P12}$ in $C1_{P9}$ and containment of $C2_{P12}$ in $C2_{P9}$	$overlap_{P9,P12}(C1) = \frac{com_{P12}(C1)}{com_{P9}(C1)}$ $overlap_{P12,P9}(C1) = 1$ $overlap_{P9,P12}(C2) = \frac{com_{P12}(C2)}{com_{P9}(C2)}$ $overlap_{P12,P9}(C2) = 1$	$overlap_{P9,P12}(r1) = \frac{com_{P12}(r1)}{com_{P9}(r1)}$ $overlap_{P12,P9}(r1) = 1$

Table 4.5 : Overlap of two peers exporting the same property with the same domain /range

By replacing in formulae (4.20), (4.21) and (4.23) the overlap in each case of Table 4.5, we get the coverage, density and completeness of two peers with respect to a property in Table 4.6:

Overlap of C1 and C2 in peers P9, P12	**Quality Metrics Estimation Formulae for property r1**
Disjointness of C1, C2	$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1)$ $den_{P9,P12}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12}}$ $com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1)$
Disjointness of C1 and independence of C2	$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1)$ $den_{P9,P12}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12}}$ $com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1)$
Disjointness of C1 and quantified overlap of C2	$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1)$ $den_{P9,P12}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12}}$ $com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1)$

Disjointness of C1 and containment of $C2_{P12}$ in $C2_{P9}$	$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1)$ $den_{P9,P12}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12}}$ $com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1)$
Independence of C1, C2	$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1) - com_{P9}(C1) * com_{P12}(C1) * com_{P9}(C2) * com_{P12}(C2)$ $den_{P9,P12}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12} - com_{P9}(r1) * com_{P12}(r1) * \|C1\| * \|C2\|}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12} - com_{P9}(C1) * com_{P12}(C1) * com_{P9}(C2) * com_{P12}(C2) * \|C1\| * \|C2\|}$ $com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1) - com_{P9}(r1) * com_{P12}(r1)$
Containment of $C1_{P12}$ in $C1_{P9}$ and containment of $C2_{P12}$ in $C2_{P9}$	$cov_{P9,P12}(r1) = cov_{P9}(r1) + cov_{P12}(r1) - \frac{com_{P12}(r1)}{com_{P9}(r1)}$ $cov_{P12,P9}(r1) = cov_{P9}(r1) + cov_{P12}(r1) - 1$ $den_{P9,P12}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12} - com_{P12}(r1)/com_{P9}(r1) * \|C1\| * \|C2\|}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12} - com_{P12}(C1)/com_{P9}(C1) * com_{P12}(C2)/com_{P9}(C2) * \|C1\| * \|C2\|}$ $den_{P12,P9}(r1) = \frac{\|r1\|_{P9} + \|r1\|_{P12} - 1 * \|C1\| * \|C2\|}{\|C1\|_{P9} * \|C2\|_{P9} + \|C1\|_{P12} * \|C2\|_{P12} - 1 * 1 * \|C1\| * \|C2\|}$ $com_{P9,P12}(r1) = com_{P9}(r1) + com_{P12}(r1) - \frac{com_{P12}(r1)}{com_{P9}(r1)}$ $com_{P12,P9}(r1) = com_{P9}(r1) + com_{P12}(r1) - 1$

Table 4.6 : Quality metrics formulae for two peers exporting the same property with the same domain and range

In Table 4.6, we can see that in the case of disjointness, where there is no overlap of peers P9 and P12 with respect to property r1, the $cov_{P9,P12}(r1)$ is simply the sum of the coverage of r1 in the two peers. The same holds for the completeness formula, while the density formula is also

formed without the overlap computations. In any other overlap case of C1 and C2, the coverage, density and completeness are formulated by subtracting either the $overlap_{P9,P12}$(r1) or the product of $overlap_{P9,P12}$(C1) and $overlap_{P9,P12}$(C2). As explained earlier, in the sum of completeness probabilities of r1 in peers P9 and P12, we have counted twice the r1 instances that are common in them. We need to correct this by subtracting the probability of the common r1 instances between peers P9 and P12, in order to calculate only once those r1 instances which appear in both peers. In the case of coverage, we need to subtract the probabilities of the common C1 and C2 instances in the two peers. It is clear that in all cases in which there is some overlap of r1 in the two peers, the values of coverage, density and completeness becomes smaller as the value of overlap gets greater.

In all formulae of Table 4.6, the completeness of each peer with respect to r1 can be written as the product of its coverage and density with respect to r1 (formulae (4.19) and (4.21) respectively).

4.3.3.2 PROPERTIES WITH SUBSUMED DOMAINS/RANGES

In Figure 11 we can see that peer P1 exports property r1 with domain class C1 and range class C2 and peer P4 exports property r1 with domain class C5, subclass of C1 and range class C6, subclass of C2. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. $||C1|| * ||C2||$, since all the instances of C5 and C6 in the PDMS are included in $||C1||$ and ||C2|| respectively.

Similar to formula (4.19), the coverage formula of the union of r1 in P1 and P4 is

$$cov_{P1,P4}(r1) = \frac{|C1|_{P1} * |C2|_{P1} + |C5|_{P4} * |C6|_{P4} - Overlap_{P1,P4}(C1UC5 \times C2UC6)}{||C1|| * ||C2||} \quad \textbf{(4.24)}$$

$$\Rightarrow \; cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1) - overlap_{P1,P4}(C1UC5) * overlap_{P1,P4}(C2U6) \quad \textbf{(4.25)}$$

The idea behind the coverage formula for the union is simple: according to definition 4.1 the coverage of property r1 in peers P1 and P4 is the ratio of the maximum possible number of

distinct r1 instances in the two peers (i.e. the sum of the product of the C1/C5 and C2/C6 instances in the two peers if we subtract their overlap of C1/C5 and C2/C6) to the maximum possible number of r1 instances in the PDMS (i.e. the product of the total C1 and C2 instances in the PDMS).

The formula for density of r1 in peers P1 and P4 will be

$$den_{P1,P4}(r1) = \frac{|r1|_{P1} + |r1|_{P4} - Overlap_{P1,P4}(r1)}{|C1|_{P1} * |C2|_{P1} + |C5|_{P4} * |C6|_{P4} - Overlap_{P1,P4}(C1UC5 \times C2UC6)} \quad \textbf{(4.26)}$$

Formula (4.26) stems from definition 4.2 : density of peers P1 and P4 with respect to property r1 is the ratio of the distinct number of r1 instances in the two peers (i.e. the sum of their r1 instances if we subtract their overlap with respect to r1) to the maximum possible number of distinct r1 instances in P1 and P4.

As far as completeness of r1 in peers P1 and P4 is concerned, we have:

$$com_{P1,P4}(r1) = cov_{P1,P4}(r1) * den_{P1,P4}(r1) = \frac{|r1|_{P1} + |r1|_{P4} - Overlap_{P1,P4}(r1)}{||C1|| * ||C2||} \quad \textbf{(4.27)} \Rightarrow$$

$$com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1) - overlap_{P1,P4}(r1) \quad \textbf{(4.28)}$$

As we can see, the completeness formula is formed as the product of coverage and density and as stated by definition 4.3 it is the ratio of the number of distinct r1 instances in peers P1 and P4 to the total maximum number of r1 instances in the PDMS.

Table 4.7 shows the overlap cases presented in Table 3.5 with respect to the the completeness metrics formulae (4.4) and (4.7):

Overlap of C1 and C2 in peers P1, P4	**Overlap Estimation Formulae for C1/C5, C2/C6**	**Overlap Estimation Formulae for property r1**

Disjointness of C1/C5, C2/C6	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = 0$	$overlap_{P1,P4}(r1) = 0$
Disjointness of C1/C5 and independence of C2/C6	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = com_{P1}(C2) * com_{P4}(C6)$	$overlap_{P1,P4}(r1) = 0$
Disjointness of C1/C5 and quantified overlap of C2/C6	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = X$, where X is a known probability	$overlap_{P1,P4}(r1) = 0$
Disjointness of C1/C5 and containment of $C6_{P4}$ in $C2_{P1}$	$overlap_{P1,P4}(C1UC5) = 0$ $overlap_{P1,P4}(C2UC6) = \frac{com_{P4}(C6)}{com_{P1}(C2)}$ $overlap_{P4,P1}(C2UC6) = 1$	$overlap_{P1,P4}(r1) = 0$
Independence of C1/C5, C2/C6	$overlap_{P1,P4}(C1UC5) = com_{P1}(C1) * com_{P4}(C5)$ $overlap_{P1,P4}(C2UC6) = com_{P1}(C2) * com_{P4}(C6)$	$overlap_{P1,P4}(r1) = com_{P1}(r1) * com_{P4}(r1)$
Containment of $C5_{P4}$ in $C1_{P1}$ and containment of $C6_{P4}$ in $C2_{P1}$	$overlap_{P1,P4}(C1UC5) = \frac{com_{P4}(C5)}{com_{P1}(C1)}$ $overlap_{P4,P1}(C1UC5) = 1$ $overlap_{P1,P4}(C2UC6) = \frac{com_{P4}(C6)}{com_{P1}(C2)}$ $overlap_{P4,P1}(C2UC6) = 1$	$overlap_{P1,P4}(r1) = \frac{com_{P4}(r1)}{com_{P1}(r1)}$ $overlap_{P4,P1}(r1) = 1$

Table 4.7 : Overlap of two peers exporting the same property with subsumed domains/ranges

By replacing in formulae (4.25), (4.26) and (4.28) the overlap in each case of Table 4.7, we get the coverage, density and completeness of two peers with respect to a property in Table 4.8:

Overlap of C1/C5 and C2/C6 in peers P1, P8	**Quality Metrics Estimation Formulae for property r1**
Disjointness of C1/C5, C2/C6	$cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1)$ $den_{P1,P4}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4}}$ $com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1)$
Disjointness of C1/C5 and independence of C2/C6	$cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1)$ $den_{P1,P4}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4}}$ $com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1)$
Disjointness of C1/C5 and quantified overlap of C2/C6	$cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1)$ $den_{P1,P4}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4}}$ $com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1)$
Disjointness of C1/C5 and containment of $C6_{P4}$ in $C2_{P1}$	$cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1)$ $den_{P1,P4}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4}}$ $com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1)$

Independence of C1/C5, C2/C6	$cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1) - com_{P1}(C1) * com_{P4}(C5) * com_{P1}(C2) * com_{P4}(C6)$ $den_{P1,P4}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4} - com_{P1}(r1) * com_{P4}(r1) * \|C1\| * \|C2\|}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4} - com_{P1}(C1) * com_{P4}(C5) * com_{P1}(C2) * com_{P4}(C6) * \|C1\| * \|C2\|}$ $com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1) - com_{P1}(r1) * com_{P4}(r1)$
Containment of $C5_{P4}$ in $C1_{P1}$ and containment of $C6_{P4}$ in $C2_{P1}$	$cov_{P1,P4}(r1) = cov_{P1}(r1) + cov_{P4}(r1) - \frac{com_{P4}(r1)}{com_{P1}(r1)}$ $cov_{P4,P1}(r1) = cov_{P1}(r1) + cov_{P4}(r1) - 1$ $den_{P1,P4}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4} - com_{P4}(r1)/com_{P1}(r1) * \|C1\| * \|C2\|}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4} - com_{P4}(C5)/com_{P1}(C1) * com_{P4}(C6)/com_{P1}(C2) * \|C1\| * \|C2\|}$ $den_{P4,P1}(r1) = \frac{\|r1\|_{P1} + \|r1\|_{P4} - 1 * \|C1\| * \|C2\|}{\|C1\|_{P1} * \|C2\|_{P1} + \|C5\|_{P4} * \|C6\|_{P4} - 1 * 1 * \|C1\| * \|C2\|}$ $com_{P1,P4}(r1) = com_{P1}(r1) + com_{P4}(r1) - \frac{com_{P4}(r1)}{com_{P1}(r1)}$ $com_{P4,P1}(r1) = com_{P1}(r1) + com_{P4}(r1) - 1$

Table 4.8 : Quality metrics formulae for two peers exporting the same property with the same domain and range

In Table 4.8, we can see that in the case of disjointness, where there is no overlap of peers P1 and P4 with respect to property r1, the $cov_{P1,P4}(r1)$ is simply the sum of the coverage of r1 in the two peers. The same holds for the completeness formula, while the density formula is also formed without the overlap computations. In any other overlap case of C1/C5 and C2/C6, the coverage, density and completeness are formulated by subtracting either the $overlap_{P1,P4}(r1)$ or the product of $overlap_{P1,P4}(C1UC5)$ and $overlap_{P1,P4}(C2UC6)$. As explained earlier, in the sum of completeness probabilities of r1 in peers P1 and P4, we have counted twice the r1 instances that are common in them. We need to correct this by subtracting the probability of the common r1 instances between peers P1 and P4, in order to calculate only once those r1 instances which

appear in both peers. In the case of coverage, we need to subtract the probabilities of the common C5 and C6 instances in the two peers. It is clear that in all cases in which there is some overlap of r1 in the two peers, the values of coverage, density and completeness becomes smaller as the value of overlap gets greater.

In all formulae of Table 4.8, the completeness of each peer with respect to r1 can be written as the product of its coverage and density with respect to r1 (formulae (4.24) and (4.28) respectively).

4.3.4 PEERS PUBLISHING SUBSUMED PROPERTIES

When a peer exports a property and another peer exports a subproperty of it, one of the following cases may hold :

- ❖ The subproperty may have the same domain and range class with the parent property.
- ❖ One of the domain/range class of the subproperty (or both of them) may be subclass of the domain/range class of the parent property.

4.3.4.1 PROPERTIES WITH THE SAME DOMAIN/RANGE

In Figure 10 we can see that peer P1 exports property r1 with domain class C1 and range class C2 and peer P3 exports property r4 with domain class C1 and range class C2. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2||. In the coverage formula we take into account the number of common r4 instances between P1 and P3. However, since property r1 subsumes property r4, the r4 instances of peer P3 can also answer queries that refer to r1. So, the coverage formula of the union of r1 in P1 and r4 in P3 is

$$\text{cov}_{P1,P3}(r1 \cup r4) = \frac{|C1|_{P1} * |C2|_{P1} + |C1|_{P3} * |C2|_{P3} - Overlap_{P1,P3}(C1 \times C2)}{||C1|| * ||C2||} \quad \textbf{(4.29)} \quad \Rightarrow$$

$$\text{cov}_{P1,P3}(r1 \cup r4) = \text{cov}_{P1}(r1) + \text{cov}_{P3}(r4) - \text{overlap}_{P1,P3}(C1 \times C2) \quad \textbf{(4.30)}$$

The idea behind the coverage formula for the union is simple: according to definition 4.1 the coverage of properties r1 and r4 in peers P1 and P3 is the ratio of the maximum possible number

of distinct r1/r4 instances in the two peers (i.e. the sum of the product of the C1 and C2 instances in the two peers if we subtract their overlap of C1 and C2 to the maximum possible number of r1 instances in the PDMS (i.e. the product of the total C1 and C2 instances in the PDMS).

The formula for density of r1 and r4 in peers P1 and P3 will be

$$\text{den}_{P1,P3}(r1Ur4)= \frac{|r1|_{P1}+|r4|_{P3}-Overlap_{P1,P3}(r1Ur4)}{|C1|_{P1}*|C2|_{P1}+|C1|_{P3}*|C2|_{P3}-Overlap_{P1,P3}(C1\times C2)} \quad \textbf{(4.31)}$$

Formula (4.31) stems from definition 4.2 : density of peers P1 and P3 with respect to properties r1 and r4 is the ratio of the distinct number of r1/r4 instances in the two peers (i.e. the sum of their r1/r4 instances if we subtract their overlap with respect to r4) to the maximum possible number of distinct r1 instances in P1 and P3.

As far as completeness of r1/r4 in peers P1 and P3 is concerned, we have:

$$\text{com}_{P1,P3}(r1Ur4)= \frac{|r1|_{P1}+|r4|_{P3}-Overlap_{P1,P3}(r1Ur4)}{||C1||*||C2||} \quad \textbf{(4.32)} \quad =$$

$= \text{cov}_{P1,P3}(r1Ur4) * \text{den}_{P1,P3}(r1Ur4) \quad \Rightarrow$

$$\text{com}_{P1,P3}(r1Ur4) = \text{com}_{P1}(r1) + \text{com}_{P3}(r4) - \text{overlap}_{P1,P3}(r1Ur4) \quad \textbf{(4.33)}$$

As we can see, the completeness formula is formed as the product of coverage and density and as stated by definition 4.3 it is the ratio of the number of distinct r1/r4 instances in peers P1 and P3 to the total maximum number of r1 instances in the PDMS.

Table 4.9 shows the overlap cases presented in Table 3.6 with respect to the the completeness metrics formulae (4.4) and (4.7):

Overlap of C1 and C2 in peers P1, P4	**Overlap Estimation Formulae for C1, C2**	**Overlap Estimation Formulae for property r1/r4**
Disjointness of C1, C2	$\text{overlap}_{P1,P3}(C1) = 0$ $\text{overlap}_{P1,P3}(C2) = 0$	$\text{overlap}_{P1,P3}(r1Ur4) = 0$

Disjointness of C1 and independence of C2	$overlap_{P1,P3}(C1) = 0$ $overlap_{P1,P3}(C2) =$ $com_{P1}(C2) * com_{P3}(C2)$	$overlap_{P1,P3}(r1Ur4) = 0$
Disjointness of C1 and quantified overlap of C2	$overlap_{P1,P4}(C1) = 0$ $overlap_{P1,P4}(C2) = X$, where X is a known probability	$overlap_{P1,P3}(r1Ur4) = 0$
Disjointness of C1 and containment of $C2_{P3}$ in $C2_{P1}$	$overlap_{P1,P3}(C1) = 0$ $overlap_{P1,P3}(C2) = \frac{com_{P3}(C2)}{com_{P1}(C2)}$ $overlap_{P3,P1}(C2) = 1$	$overlap_{P1,P3}(r1Ur4) = 0$
Independence of C1, C2	$overlap_{P1,P3}(C1) =$ $com_{P1}(C1) * com_{P3}(C1)$ $overlap_{P1,P3}(C2) =$ $com_{P1}(C2) * com_{P3}(C2)$	$overlap_{P1,P3}(r1Ur4) =$ $com_{P1}(r1) * com_{P3}(r4)$
Containment of $C1_{P3}$ in $C1_{P1}$ and containment of $C2_{P3}$ in $C2_{P1}$	$overlap_{P1,P3}(C1) = \frac{com_{P3}(C1)}{com_{P1}(C1)}$ $overlap_{P3,P1}(C1) = 1$ $overlap_{P1,P3}(C2) = \frac{com_{P3}(C2)}{com_{P1}(C2)}$ $overlap_{P3,P1}(C2) = 1$	$overlap_{P1,P3}(r1Ur4) = \frac{com_{P3}(r4)}{com_{P1}(r1)}$ $overlap_{P3,P1}(r1Ur4) = 1$

Table 4.9 :Overlap of two peers exporting subsumed properties with the same domain/range

By replacing in formulae (4.30), (4.31) and (4.33) the overlap in each case of Table 4.9, we get the coverage, density and completeness of two peers with respect to subsumed properties in Table 4.10:

Overlap of C1 and C2 in peers P1, P3	**Quality Metrics Estimation Formulae for properties r1/r4**
Disjointness of C1, C2	$cov_{P1,P3}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4)$ $den_{P1,P3}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3}}$ $com_{P1,P3}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4)$
Disjointness of C1 and independence of C2	$cov_{P1,P3}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4)$ $den_{P1,P3}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3}}$ $com_{P1,P3}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4)$
Disjointness of C1 and quantified overlap of C2	$cov_{P1,P3}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4)$ $den_{P1,P3}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3}}$ $com_{P1,P3}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4)$
Disjointness of C1 and containment of $C2_{P3}$ in $C2_{P1}$	$cov_{P1,P3}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4)$ $den_{P1,P3}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3}}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3}}$ $com_{P1,P3}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4)$

Independence of C1, C2	$cov_{P1,P3}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4) - com_{P1}(C1) * com_{P3}(C1) * com_{P1}(C2) * com_{P3}(C2)$ $den_{P1,P3}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3} - com_{P1}(r1) * com_{P3}(r4) * \|C1\| * \|C2\|}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3} - com_{P1}(C1) * com_{P3}(C1) * com_{P1}(C2) * com_{P3}(C2) * \|C1\| * \|C2\|}$ $com_{P1,P3}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4) - com_{P1}(r1) * com_{P3}(r4)$
Containment of $C1_{P3}$ in $C1_{P1}$ and containment of $C2_{P3}$ in $C2_{P1}$	$cov_{P1,P3}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4) - \frac{com_{P3}(r4)}{com_{P1}(r1)}$ $cov_{P3,P1}(r1Ur4) = cov_{P1}(r1) + cov_{P3}(r4) - 1$ $den_{P1,P3}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3} - com_{P3}(r4)/com_{P1}(r1) * \|C1\| * \|C2\|}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3} - com_{P3}(C1)/com_{P1}(C1) * com_{P3}(C2)/com_{P1}(C2) * \|C1\| * \|C2\|}$ $den_{P3,P1}(r1Ur4) = \frac{\|r1\|_{P1} + \|r4\|_{P3} - 1 * \|C1\| * \|C2\|}{\|C1\|_{P1} * \|C2\|_{P1} + \|C1\|_{P3} * \|C2\|_{P3} - 1 * 1 * \|C1\| * \|C2\|}$ $com_{P1,P3}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4) - \frac{com_{P3}(r4)}{com_{P1}(r1)}$ $com_{P3,P1}(r1Ur4) = com_{P1}(r1) + com_{P3}(r4) - 1$

Table 4.10 : Quality metrics formulae for two peers exporting subsumed properties with the same domain and range

In Table 4.10, we can see that in the case of disjointness, where there is no overlap of peers P1 and P3 with respect to properties r1/r4, the $cov_{P1,P3}(r1Ur4)$ is simply the sum of the coverage of r1/r4 in the two peers. The same holds for the completeness formula, while the density formula is also formed without the overlap computations. In any other overlap case of C1 and C2, the coverage, density and completeness are formulated by subtracting either the $overlap_{P1,P3}(r1Ur4)$ or the product of $overlap_{P1,P3}(C1)$ and $overlap_{P1,P3}(C2)$. As explained earlier, in the sum of completeness probabilities of r1/r4 in peers P1 and P3, we have counted twice the r4 instances that are common in them. We need to correct this by subtracting the probability of the common r4 instances between peers P1 and P3, in order to calculate only once those r4 instances which

appear in both peers. In the case of coverage, we need to subtract the probabilities of the common C1 and C2 instances in the two peers. It is clear that in all cases in which there is some overlap of r1/r4 in the two peers, the values of coverage, density and completeness becomes smaller as the value of overlap gets greater.

In all formulae of Table 4.10, the completeness of each peer with respect to r1/r4 can be written as the product of its coverage and density with respect to r1/r4 (formulae (4.29) and (4.31) respectively).

4.3.4.2 PROPERTIES WITH SUBSUMED DOMAINS/RANGES

In Figure 3 we can see that peer P9 exports property r1 with domain class C1 and range class C2 and peer P13 exports property r4, which is a subclass of r1 with domain class C7, subclass of C1 and range class C6, subclass of C2. Since r1 subsumes r4, the r4 instances of P13 can answer not only queries concerning r4, but also queries concerning r1. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e.||C1|| * ||C2||, since all the instances of C7 and C6 in the PDMS are included in ||C1|| and ||C2|| respectively.

The coverage formula of the union of r1 and r4 in P9 and P13 is

$$\text{cov}_{P9,P13}(r1 \cup r4) = \frac{|C1|_{P9} * |C2|_{P9} + |C7|_{P13} * |C6|_{P13} - Overlap_{P9,P13}(C1 \cup C7 \times C2 \cup C6)}{||C1|| * ||C2||} \quad \textbf{(4.34)}$$

=>

$$\text{cov}_{P9,P13}(r1 \cup r4) = \text{cov}_{P9}(r1) + \text{cov}_{P13}(r4) - \text{overlap}_{P9,P13}(C1 \cup C7) * \text{overlap}_{P9,P13}(C2 \cup 6) \quad \textbf{(4.35)}$$

The idea behind the coverage formula for the union is simple: according to definition 4.1 the coverage of properties r1/r4 in peers P9 and P13 is the ratio of the maximum possible number of distinct r4 instances in the two peers (i.e. the sum of the product of the C1/C7 and C2/C6 instances in the two peers if we subtract their overlap of C1/C7 and C2/C6) to the maximum possible number of r1 instances in the PDMS (i.e. the product of the total C1 and C2 instances in the PDMS).

The formula for density of r1/r4 in peers P9 and P13 will be

$$\text{den}_{P9,P13}(r1 \cup r4) = \frac{|r1|_{P9} + |r4|_{P13} - Overlap_{P9,P13}(r1 \cup r4)}{|C1|_{P9} * |C2|_{P9} + |C7|_{P13} * |C6|_{P13} - Overlap_{P9,P13}(C1 \cup C7 \times C2 \cup C6)} \quad \textbf{(4.36)}$$

Formula (4.36) stems from definition 4.2 : density of peers P9 and P13 with respect to properties r1 and r4 is the ratio of the distinct number of r4 instances in the two peers (i.e. the sum of their r1/r4 instances if we subtract their overlap with respect to r4) to the maximum possible number of distinct r1/r4 instances in P9 and P13.

As far as completeness of r1/r4 in peers P9 and P13 is concerned, we have:

$$com_{P9,P13}(r1Ur4)=cov_{P9,P13}(r1Ur4)*den_{P9,P13}(r1Ur4)=\frac{|r1|_{P9}+|r4|_{P13}-Overlap_{P9,P13}(r1Ur4)}{\|C1\|*\|C2\|} \quad \textbf{(4.37)}$$

=>

$$com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4) - overlap_{P9,P13}(r1Ur4) \quad \textbf{(4.38)}$$

As we can see, the completeness formula is formed as the product of coverage and density and as stated by definition 4.3 it is the ratio of the number of distinct r1/r4 instances in peers P9 and P13 to the total maximum number of r1 instances in the PDMS.

Table 4.11 shows the overlap cases presented in Table 3.7 with respect to the completeness metrics formulae (4.4) and (4.7):

Overlap of C1 and C2 in peers P9, P13	**Overlap Estimation Formulae for C1/C7, C2/C6**	**Overlap Estimation Formulae for property r1/r4**
Disjointness of C1/C7, C2/C6	$overlap_{P9,P13}(C1UC7) = 0$ $overlap_{P9,P13}(C2UC6) = 0$	$overlap_{P9,P13}(r1Ur4) = 0$
Disjointness of C1/C7 and independence of C2/C6	$overlap_{P9,P13}(C1UC7) = 0$ $overlap_{P9,P13}(C2UC6)=$ $com_{P9}(C2) * com_{P13}(C6)$	$overlap_{P9,P13}(r1Ur4) = 0$
Disjointness of C1/C7 and quantified overlap of C2/C6	$overlap_{P9,P13}(C1UC7) = 0$ $overlap_{P9,P13}(C2UC6) = X$, where X is a known probability	$overlap_{P9,P13}(r1Ur4) = 0$

Disjointness of C1/C7 and containment of $C6_{P13}$ in $C2_{P9}$	$overlap_{P9,P13}(C1UC7) = 0$ $overlap_{P9,P13}(C2UC6) = \frac{com_{P13}(C6)}{com_{P9}(C2)}$ $overlap_{P13,P9}(C2UC6) = 1$	$overlap_{P9,P13}(r1Ur4) = 0$
Independence of C1/C7, C2/C6	$overlap_{P9,P13}(C1UC7) = com_{P9}(C1) * com_{P13}(C5)$ $overlap_{P9,P13}(C2UC6) = com_{P9}(C2) * com_{P13}(C6)$	$overlap_{P9,P13}(r1Ur4) = com_{P9}(r1) * com_{P13}(r4)$
Containment of $C7_{P13}$ in $C1_{P9}$ and containment of $C6_{P13}$ in $C2_{P9}$	$overlap_{P9,P13}(C1UC7) = \frac{com_{P13}(C7)}{com_{P9}(C1)}$ $overlap_{P13,P9}(C1UC7) = 1$ $overlap_{P9,P13}(C2UC6) = \frac{com_{P13}(C6)}{com_{P9}(C2)}$ $overlap_{P13,P9}(C2UC6) = 1$	$overlap_{P9,P13}(r1Ur4) = \frac{com_{P13}(r4)}{com_{P9}(r1)}$ $overlap_{P13,P9}(r1Ur4) = 1$

Table 4.11 : Overlap of two peers exporting the same property with subsumed domains/ranges

By replacing in formulae (4.25), (4.26) and (4.28) the overlap in each case of Table 4.11, we get the coverage, density and completeness of two peers with respect to a property in Table 4.12:

Overlap of C1/C7 and C2/C6 in peers P9, P13	**Quality Metrics Estimation Formulae for properties r1/r4**
Disjointness of C1/C7, C2/C6	$cov_{P9,P13}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4)$ $den_{P9,P13}(r1Ur4) = \frac{\|r1\|_{P9} + \|r4\|_{P13}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C7\|_{P13} * \|C6\|_{P13}}$ $com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4)$

Disjointness of C1/C7 and independence of C2/C6	$cov_{P9,P13}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4)$ $den_{P9,P13}(r1Ur4) = \frac{\|r1\|_{P9} + \|r4\|_{P13}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C7\|_{P13} * \|C6\|_{P13}}$ $com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4)$
Disjointness of C1/C7 and quantified overlap of C2/C6	$cov_{P9,P13}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4)$ $den_{P9,P13}(r1Ur4) = \frac{\|r1\|_{P9} + \|r4\|_{P13}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C7\|_{P13} * \|C6\|_{P13}}$ $com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4)$
Disjointness of C1/C7 and containment of $C6_{P13}$ in $C2_{P9}$	$cov_{P9,P13}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4)$ $den_{P9,P13}(r1Ur4) = \frac{\|r1\|_{P9} + \|r4\|_{P13}}{\|C1\|_{P9} * \|C2\|_{P9} + \|C7\|_{P13} * \|C6\|_{P13}}$ $com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4)$
Independence of C1/C7, C2/C6	$cov_{P9,P13}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4) - com_{P9}(C1) * com_{P13}(C7) * com_{P9}(C2) * com_{P13}(C6)$ $den_{P9,P13}(r1Ur4) = \frac{\|r1\|_{P9} + \|r4\|_{P13} - com_{P9}(r1) * com_{P13}(r4) * \|C1\| * \|C2\|}{\|C1\|_{P9} * \|C2\|_{P9} + \|C7\|_{P13} * \|C6\|_{P13} - com_{P9}(C1) * com_{P13}(C7) * com_{P9}(C2) * com_{P13}(C6) * \|C1\| * \|C2\|}$ $com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4) - com_{P9}(r1) * com_{P13}(r4)$

Containment of $C7_{P13}$ in $C1_{P9}$ and containment of $C6_{P13}$ in $C2_{P9}$	$cov_{P9,P13}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4) - \frac{com_{P13}(r4)}{com_{P9}(r1)}$ $cov_{P13,P9}(r1Ur4) = cov_{P9}(r1) + cov_{P13}(r4) - 1$ $den_{P9,P13}(r1Ur4) = \frac{\lvert r1\rvert_{P9} + \lvert r4\rvert_{P13} - com_{P13}(r4)/com_{P9}(r1) * \lVert C1\rVert * \lVert C2\rVert}{\lvert C1\rvert_{P9} * \lvert C2\rvert_{P9} + \lvert C7\rvert_{P13} * \lvert C6\rvert_{P13} - com_{P13}(C7)/com_{P9}(C1) * com_{P13}(C6)/com_{P9}(C2) * \lVert C1\rVert * \lVert C2\rVert}$ $den_{P13,P9}(r1Ur4) = \frac{\lvert r1\rvert_{P9} + \lvert r4\rvert_{P13} - 1 * \lVert C1\rVert * \lVert C2\rVert}{\lvert C1\rvert_{P9} * \lvert C2\rvert_{P9} + \lvert C7\rvert_{P13} * \lvert C6\rvert_{P13} - 1 * 1 * \lVert C1\rVert * \lVert C2\rVert}$ $com_{P9,P13}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4) - \frac{com_{P13}(r4)}{com_{P9}(r1)}$ $com_{P13,P9}(r1Ur4) = com_{P9}(r1) + com_{P13}(r4) - 1$

Table 4.12 : Quality metrics formulae for two peers exporting the same property with the same domain and range

In Table 4.12, we can see that in the case of disjointness, where there is no overlap of peers P9 and P13 with respect to properties r1 and r4, $cov_{P9,P13}(r1Ur4)$ is simply the sum of the coverage of r1 and r4 in the two peers. The same holds for the completeness formula, while the density formula is also formed without the overlap computations. In any other overlap case of C1/C7 and C2/C6, the coverage, density and completeness are formulated by subtracting either the $overlap_{P9,P13}(r1Ur4)$ or the product of $overlap_{P9,P13}(C1UC7)$ and $overlap_{P9,P13}(C2UC6)$. As explained earlier, in the sum of completeness probabilities of r1/r4 in peers P9 and P13, we have counted twice the r4 instances that are common in them. We need to correct this by subtracting the probability of the common r4 instances between peers P9 and P13, in order to calculate only once those r4 instances which appear in both peers. In the case of coverage, we need to subtract the probabilities of the common C7 and C6 instances in the two peers. It is clear that in all cases in which there is some overlap of r4 in the two peers, the values of coverage, density and completeness become smaller as the value of overlap gets greater.

In all formulae of Table 4.12, the completeness of each peer with respect to r1/r4 can be written as the product of its coverage and density with respect to r1/r4 (formulae (4.24) and (4.28) respectively).

4.4 GENERALIZATION FOR MORE THAN TWO PEERS

In section 3.6 we stated that, in favor of simplicity, when it comes to more than two peers we consider only independence overlap for them. In this section we will show how the quality metrics formulae of the previous sections are formed in the case of more than two peers.

First of all we will take the case where more than two peers export the same property. Then we have the following cases :

- ✓ The common property may have the same domain and range classes in all of the peers.
- ✓ The common property may have a specific domain/range class in some peers and subclasses of them as domain/range classes in some other peers.

Let us assume that peers P1, P2,…,PN export property r1 with the same domain and range classes, i.e. classes C1 and C2 respectively. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2|| .

Then, taking formula (4.19) into account, we have

$$\text{cov}_{P1,P2,\ldots,PN}(r1)=\frac{|C1|_{P1}*|C2|_{P1}+\ldots+|C1|_{PN}*|C2|_{PN}-Overlap_{P1,P2\ldots PN}(C1\times C2)}{||C1||*||C2||} \quad \textbf{(4.39)}$$

The density formula (4.21) will become

$$\text{den}_{P1,P2,\ldots,PN}(r1)=\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r1|_{PN}-Overlap_{P1,P2\ldots PN}(r1)}{|C1|_{P1}*|C2|_{P1}+\ldots+|C1|_{PN}*|C2|_{PN}-Overlap_{P1,P2,\ldots,PN}(C1\times C2)} \quad \textbf{(4.40)}$$

As for completeness, taking formula (4.22) into account we have :

$$\text{com}_{P1,P2,\ldots,PN}(r1) = \text{cov}_{P1,P2,\ldots,PN}(r1) * \text{den}_{P1,P2,\ldots,PN}(r1) =$$

$$=\frac{|r1|_{P1}+|r1|_{P2}+\ldots+|r1|_{PN}-Overlap_{P1,P2\ldots PN}(r1)}{||C1||*||C2||} \quad \textbf{(4.41)}$$

The other case of peers exporting the same property is that some, or all of them may have as domain a subclass of C1 (e.g. C5) and as range class a subclass of C2, e.g. (C6). Let us assume that peer P1 exports property r1 with C1 as domain class and C2 as range class, and all other

peers export r1 with domain class C5 and range class C6. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2||, since all the instances of C5 and C6 in the PDMS are included in ||C1|| and ||C2|| respectively.

Then, the coverage formula (4.39) becomes

$$\text{cov}_{P1,P2,\ldots,PN}(r1)=\frac{|C1|_{P1}*|C2|_{P1}+\ldots+|C5|_{PN}*|C6|_{PN}-Overlap_{P1,P2\ldots PN}(C1UC5\times C2UC6)}{||C1||*||C2||} \quad \textbf{(4.42)}$$

The density formula (4.26) will become

$$\text{den}_{P1,P2,\ldots,PN}(r1)=\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r1|_{PN}-Overlap_{P1,P2\ldots PN}(r1)}{|C1|_{P1}*|C2|_{P1}+\ldots+|C5|_{PN}*|C6|_{PN}-Overlap_{P1,P2,\ldots,PN}(C1UC5\times C2UC6)} \quad \textbf{(4.43)}$$

As for completeness, the formula is the same as (4.41).

Since we consider only independence overlap among the peers, then, using the overlap information of Table 3.4 and Table 3.5, the quality metrics formulae (4.20), (4.21), (4.23), (4.25) and (4.28) are shown in Table 4.13.

Same property (same domain and range classes)
$\text{cov}_{P1,\ldots PN}(r1) = \text{cov}_{P1}(r1)+\ldots+\text{cov}_{PN}(r1) -\text{com}_{P1}(C1)*\ldots*\text{com}_{PN}(C1) *_{\ldots}*\text{com}_{p1}(C2)*\text{com}_{PN}(C2)$ $\text{den}_{P1,P2,\ldots,PN}(r1)=$ $\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r1|_{PN}-com_{P1}(r1)*\ldots*com_{PN}(r1)*\|C1\|*\|C2\|}{|C1|_{P1}*|C2|_{P1}+\ldots+|C1|_{PN}*|C2|_{PN}-com_{P1}(C1)*com_{PN}(C1)*\ldots*com_{P1}(C2)*com_{PN}(C2)*\|C1\|*\|C2\|}$ $\text{com}_{P1,\ldots PN}(r1) = \text{com}_{P1}(r1)+\ldots+\text{com}_{PN}(r1) - \text{com}_{P1}(r1)* \ldots *\text{com}_{PN}(r1)$
Same property (subsumed domain/range classes)
$\text{cov}_{P1,\ldots PN}(r1) = \text{cov}_{P1}(r1)+\ldots+\text{cov}_{PN}(r1) - \text{com}_{P1}(C1)*\ldots*\text{com}_{P1}(C2) * \ldots *\text{com}_{pN}(C5)*\text{com}_{PN}(C6)$ $\text{den}_{P1,P2,\ldots,PN}(r1)=$ $\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r1|_{PN}-com_{P1}(r1)*\ldots*com_{PN}(r1)*\|C1\|*\|C2\|}{|C1|_{P1}*|C2|_{P1}+\ldots+|C5|_{PN}*|C6|_{PN}-com_{P1}(C1)*com_{P1}(C2)*\ldots*com_{PN}(C5)*com_{PN}(C6)*\|C1\|*\|C2\|}$ $\text{com}_{P1,\ldots PN}(r1) = \text{com}_{P1}(r1)+\ldots+\text{com}_{PN}(r1) - \text{com}_{P1}(r1)* \ldots *\text{com}_{PN}(r1)$

Table 4.13 : Quality metrics formulae for more than two peers exporting the same property

When there are peers that export e.g. property r1 and there are also peers that export a subproperty of r1, e.g. r4, then there are two cases :

- ✓ Both property and subproperty have the same domain and range classes.
- ✓ One or both of the domain/range class of the subproperty is a subclass of the domain/range class of the property.

Let us assume that peers P1, P2, …,PK export property r1 with domain class C1 and range class C2, while peers P(K+1) … PN export property r4, a subproperty of r1 with the same domain and range classes. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2||.

Then, taking formula (4.29) into account, we have

$$\text{cov}_{P1,P2,\ldots,PN}(r1 \cup r4) = \frac{|C1|_{P1} * |C2|_{P1} + .. + |C1|_{PN} * |C2|_{PN} - Overlap_{P1,P2\ldots PN}(C1 \times C2)}{||C1|| * ||C2||} \quad \textbf{(4.44)}$$

The density formula (4.31) will then become

$$\text{den}_{P1,P2,\ldots,PN}(r1) = \frac{|r1|_{P1} + |r1|_{P2} \ldots + |r4|_{PN} - com_{P1}(r1) * \ldots * com_{PN}(r4) * ||C1|| * ||C2||}{|C1|_{P1} * |C2|_{P1} + \ldots + |C1|_{PN} * |C2|_{PN} - com_{P1}(C1) * com_{P1}(C2) * \ldots * com_{PN}(C1) * com_{PN}(C2) * ||C1|| * ||C2||} \quad \textbf{(4.45)}$$

The completeness formula, taking into account formula (4.32) will be

$$\text{com}_{P1,P2,\ldots,PN}(r1 \cup r4) = \text{cov}_{P1,P2,\ldots,PN}(r1 \cup r4) * \text{den}_{P1,P2,\ldots,PN}(r1 \cup r4) =$$

$$= \frac{|r1|_{P1} + |r1|_{P2} + \ldots + |r4|_{PN} - Overlap_{P1,P2\ldots PN}(r1 \cup r4)}{||C1|| * ||C2||} \quad \textbf{(4.46)}$$

The other case is that some peers, e.g. peers P1 … PK, may export property r1 with domain class C1 and range class C2, while peers P(K+1) … PN may export subproperty r4 with domain/range that is a subclass of domain/range of r1, e.g. C5, subclass of C1 and C6, subclass of C2. The reference set used in the formulae is the product of the total number of C1 and C2 instances in the PDMS, i.e. ||C1|| * ||C2||, since all the instances of C5 and C6 in the PDMS are included in ||C1|| and ||C2|| respectively.

Then, coverage formula (4.44) becomes

$cov_{P1,P2,\ldots,PN}(r1Ur4)$ =

$$\frac{|C1|_{P1} * |C2|_{P1} + \ldots + |C5|_{PN} * |C6|_{PN} - Overlap_{P1,\,P2\ldots PN}(C1UC5 \times C2UC6)}{||C1|| * ||C2||} =$$

$cov_{P1}(r1)+\ldots+cov_{PN}(r4)$ – $overlap_{P1,P2,\ldots,pk}(C1)$* $overlap_{P1,P2,\ldots,pk}(C2)$* $overlap_{Pk+1,P2,\ldots,pN}(C5)$* $overlap_{Pk+1,P2,\ldots,pN}(C6)$ ***(4.47)***

The density formula will be

$den_{P1,P2,\ldots,PN}(r1Ur4)$=

$$\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r4|_{PN} - com_{P1}(r1)*\ldots*com_{PN}(r4)*||C1||*||C2||}{|C1|_{P1}*|C2|_{P1}+\ldots+|C5|_{PN}*|C6|_{PN} - com_{P1}(C1)*com_{P1}(C2)*\ldots*com_{PN}(C5)*com_{PN}(C6)*||C1||*||C2||} \quad \textbf{(4.48)}$$

As for completeness, the formula is the same as (4.46).

Since we consider only independence overlap among the peers, then, using the overlap information of Table 3.6 and Table 3.7, the quality metrics formulae (4.30), (4.31), (4.33), (4.35), (4.36) and (4.38) are shown in Table 4.14

Subsumed properties (same domain and range classes)
$cov_{P1,\ldots PN}(r1Ur4) = cov_{P1}(r1)+\ldots+cov_{PN}(r4) - com_{P1}(C1)*com_{P1}(C2)*_{,\ldots,}*com_{pN}(C1)*com_{PN}(C2)$ $den_{P1,P2,\ldots,PN}(r1Ur4)$= $\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r4|_{PN} - com_{P1}(r1)*\ldots*com_{PN}(r4)*\|C1\|*\|C2\|}{\|C1\|_{P1}*\|C2\|_{P1}+\ldots+\|C1\|_{PN}*\|C2\|_{PN} - com_{P1}(C1)*com_{P1}(C2)*\ldots*com_{PN}(C1)*com_{PN}(C2)*\|C1\|*\|C2\|}$ $com_{P1,\ldots PN}(r1Ur4) = com_{P1}(r1)+\ldots+com_{PN}(r4) - com_{P1}(r1)* \ldots *com_{PN}(r4)$
Subsumed properties (subsumed domain/range classes)
$cov_{P1,\ldots PN}(r1Ur4) = cov_{P1}(r1)+\ldots+cov_{PN}(r4) - com_{P1}(C1)*com_{P1}(C2)*_{,\ldots,}*com_{pN}(C5)*com_{PN}(C6)$ $den_{P1,P2,\ldots,PN}(r1Ur4)$= $\frac{|r1|_{P1}+|r1|_{P2}\ldots+|r4|_{PN} - com_{P1}(r1)*\ldots*com_{PN}(r4)*\|C1\|*\|C2\|}{\|C1\|_{P1}*\|C2\|_{P1}+\ldots+\|C5\|_{PN}*\|C6\|_{PN} - com_{P1}(C1)*com_{P1}(C2)*\ldots*com_{PN}(C5)*com_{PN}(C6)*\|C1\|*\|C2\|}$ $com_{P1,\ldots PN}(r1Ur4) = com_{P1}(r1)+\ldots+com_{PN}(r4) - com_{P1}(r1)* \ldots *com_{PN}(r4)$

Table 4.14 : Quality metrics formulae for more than two peers exporting subsumed properties

4.5 JOIN OF PEER FRAGMENT INSTANCES

When a peer exports a more complex fragment than a single class or property, it involves joins among properties. There are many ways in which two or more properties can be joined with each other, which we describe in the next section.

4.5.1 DIFFERENT JOIN TYPES (CHAIN, STAR ON DOMAINS OR RANGES)

In favor of simplicity, we will first take the case of two properties joined in a peer to form a more complex fragment. We have already presented the quality metrics formulae for fragments that involve two or more properties joined together in section 4.2.3.

There are the following cases as to the morphology the join may have:

A. The class on which the two properties join may be the range class of the one property and the domain class of the other.

B. The class on which the two properties join on may be the domain class for both properties.

C. The class on which the two properties join on may be the range class for both properties.

When we have more than two properties, the first kind of join is called a chain, the second kind of join is called a star join on the domain classes of the properties and the third is called a star join on their range classes. In the following sections we will give examples of these cases, both in the case where two or more peers export fragments that involve join of two or more properties (i.e. the join takes place in the same peer) and in the case where different peers export fragments that join on their common class (i.e. the join takes place between different peers). We should note that in the following formulae where the symbol × appears, it denotes the join symbol ⋈ that we use.

4.5.1.1 PROPERTIES WITH THE SAME DOMAIN/RANGE

As shown in Figure 3, peers P12 and P14 export properties r1 and r2 with the same domain and range classes, joined on class C2. This is a chain join, since C2 is the range class of property r1 and the domain class of property r2. The reference set used in the formulae is the product of the total number of the instances of the classes that r1 ⋈ r2 comprises, i.e. ||C1|| * ||C2|| * ||C3||. Peer P9 exports the join of properties r1 and r3. This is a star join on their domain class, C1. In the figures of this thesis we do not have any example of star join of properties on their range class, but this case is similar to the star join on the domain class.

The quality metrics for the join of r1 and r2, using formulae (4.8) and (4.19), are given below:

$cov_{P12,P14}(r1 \bowtie r2)=$

$$\frac{|C1|_{P12}*|C2|_{P12}*|C3|_{P12}+|C1|_{P14}*|C2|_{P14}*|C3|_{P14}-Overlap_{P12,P14}(C1xC2xC3)}{||C1||*||C2||*||C3||} \quad \textbf{(4.49)}$$

The coverage of the union will be the ratio of the sum of the maximum number of distinct r1 ⋈ r2 instances in the two peers (this is why we subtract the common C1, C2 and C3 instances, as they appear in both peers) to the maximum total number of r1 ⋈ r2 instances in the PDMS, i.e. ||C1|| * ||C2|| * ||C3||.

In section 3.6 we presented formulae for the overlap between peers that export the same complex fragment. These overlap formulae consider the cardinality of the complex fragment in each peer. However, in the coverage formula (4.49) we need to estimate the overlap between peers P12 and P14 with respect to the maximum possible number of r1 ⋈ r2 instances in both peers. The maximum number of r1 ⋈ r2 instances in e.g. peer P12 is $|C1|_{P12}$ * $|C2|_{P12}$ * $|C3|_{P12}$. In the same way we can compute the maximum number of r1 ⋈ r2 instances of peer P14.

Then, the overlap of formula (4.49) expressed as probability in the independence case will be computed as follows :

$overlap_{P12,P14}(C1 \times C2 \times C3) = com_{P12,P14}(C1) * com_{P12,P14}(C2) * com_{P12,P14}(C3)$

Using formula (3.3), the relationship between the overlap as probability and as cardinality is :

$overlap_{P12,P14}(r1 \times r2) = Overlap_{P12,P14}(r1 \times r2) / (||C1|| *||C2|| *||C3||)$

The coverage formula (4.49) can be written as

$cov_{P12,P14}(r1 \bowtie r2) = cov_{P12}(r1 \bowtie r2) + cov_{P14}(r1 \bowtie r2) - overlap_{P12,P14}(C1) * overlap_{P12,P14}(C2) * overlap_{P12,P14}(C3)$ ***(4.50)***

The density formula is as follows :

$$den_{P12,P14}(r1 \bowtie r2) = \frac{|r1 \times r2|_{P12} + |r1 \times r2|_{P14} - Overlap_{P12,P14}(r1 \times r2)}{|C1|_{P12} * |C2|_{P12} * |C3|_{P12} + |C1|_{P14} * |C2|_{P14} * |C3|_{P14} - Overlap_{P12,P14}(C1xC2xC3)} \quad \textbf{(4.51)}$$

The overlap of the numerator is the overlap used in the completeness computations. It is estimated as follows :

$$overlap_{P12,P14}(r1 \bowtie r2) = \frac{|r1 \times r2|_{P12}}{||C1|| * ||C2|| * ||C3||} * \frac{|r1 \times r2|_{P14}}{||C1|| * ||C2|| * ||C3||}$$

As for completeness, we have:

$$com_{P12,P14}(r1 \bowtie r2) = \frac{|r1 \times r2|_{P12} + |r1 \times r2|_{P14} - Overlap_{P12,P14}(r1 \times r2)}{||C1|| * ||C2|| * ||C3||} \quad \textbf{(4.52)}$$

Using formula (3.3), the relationship between the overlap as probability and as cardinality is :

$overlap_{P12,P14}(r1 \bowtie r2) = Overlap_{P12,P14}(r1 \bowtie r2) / (||C1|| * ||C2|| * ||C3||)$

As a result, if we consider completeness formula (4.10), formula (4.52) can be written as

$com_{P12,P14}(r1 \bowtie r2) = com_{P12}(r1 \bowtie r2) + com_{P14}(r1 \bowtie r2) - overlap_{P12,P14}(r1 \bowtie r2)$ ***(4.53)***

We should point out that in formula (4.53) overlap is expressed as a probability. In section 3.7 we have shown formulae for the overlap computation of a complex peer fragment. In favor of simplicity we assume only independence overlap among the classes of the join.

4.5.1.2 PROPERTIES WITH SUBSUMED DOMAINS/RANGES

In Figure 3 we can see that peer P12 exports $r1 \bowtie r2$, with classes C1, C2 and C3 as domain/range classes for properties r1 and r2, while peer P13 exports fragment $r4 \bowtie r2$, where r4 is a subproperty of r1 and has C7, subclass of C1 as domain class and C6, subclass of C2 as range class. The reference set used in the formulae is the product of the total number of C1, C2 and C3 instances in the PDMS, i.e. ||C1|| * ||C2|| * ||C3||, since all the instances of C7 and C6 in the

PDMS are included in ||C1|| and ||C2|| respectively. In favor of simplicity we assume only independence overlap among the classes of the join. Then, similarly to formula (4.49), the formula for coverage will become as follows:

$cov_{P12,P13}((r1 \cup r4) \bowtie r2) =$

$$\frac{|C1|_{P12} * |C2|_{P12} * |C3|_{P12} + |C7|_{P13} * |C6|_{P13} * |C3|_{P13} - Overlap_{P12,P16}(C1UC7xC2UC6xC3)}{||C1|| * ||C2|| * ||C3||} \quad \textbf{(4.54)}$$

According to formula (4.50), formula (4.54) can also be written as

$cov_{P12,P13}((r1 \cup r4) \bowtie r2) = cov_{P12}(r1 \bowtie r2) + cov_{P13}(r4 \bowtie r2) -$
$overlap_{P12,P13}(C1UC7) * overlap_{P12,P13}(C2UC6) * overlap_{P12,P13}(C3)$ ***(4.55)***

The coverage of the union will be the ratio of the sum of the maximum number of distinct $r1 \bowtie r2$ instances in the two peers (this is why we subtract the overlap of (C1UC7) x (C2UC6) x C3 instances that are counted twice, as they appear in both peers) to the maximum total number of distinct $r1 \bowtie r2$ instances in the PDMS, i.e. ||C1|| * ||C2|| * ||C3||. In formula (4.55) overlap is expressed as a probability.

The density formula is similar to formula (4.51) :

$den_{P12,P13}((r1 \cup r4) \bowtie r2) =$

$$\frac{|r1 \times r2|_{P12} + |r4 \times r2|_{P13} - Overlap_{P12,P13}((r1 \cup r4) \times r2)}{|C1|_{P12} * |C2|_{P12} * |C3|_{P12} + |C7|_{P13} * |C6|_{P13} * |C3|_{P13} - Overlap_{P12,P13}(C1UC7xC2UC6xC3)} \quad \textbf{(4.56)}$$

As for completeness, similarly to formula (4.52), we have :

$$com_{P12,P13}((r1 \cup r4) \bowtie r2) = \frac{|r1 \times r2|_{P12} + |r4 \times r2|_{P13} - Overlap_{P12,P13}((r1 \cup r4) \times r2)}{||C1|| * ||C2|| * ||C3||} \quad \textbf{(4.57)}$$

According to formula (4.53), completeness formula (4.57) can be written as

$com_{P12,P13}((r1 \cup r4) \bowtie r2) = com_{P12}(r1 \bowtie r2) + com_{P13}(r4 \bowtie r2) - overlap_{P12,P13}((r1 \cup r4) \bowtie r2)$
(4.58)

4.5.1.3 GENERALIZATION FOR MORE THAN TWO PEERS

Let us assume that peers P1, P2,…,PN export r1 ⋈ r2 with the same domain and range classes. Then the coverage formula (4.50) will become :

$cov_{P1,P2,...,PN}(r1 \bowtie r2)=cov_{P1}(r1 \bowtie r2)+\ldots+cov_{PN}(r1 \bowtie r2)-overlap_{P1,...,PN}(C1)*$
$overlap_{P1,...,PN}(C2)*overlap_{P1,...,PN}(C3) = \Sigma_i cov_{Pi}(r1 \bowtie r2) - \Pi_j overlap_{Pj}(C_j)$ ***(4.59)***

where i = 1, …, N and j = 1, 2, 3.

The density formula (4.51) will become

$den_{P1,P2....,PN}(r1 \bowtie r2)=$

$$\frac{|r1\times r2|_{P1}+...+|r1\times r2|_{PN}-Overlap_{P1,P2,...,PN}(r1\times r2)}{|C1|_{P1}*|C2|_{P1}*|C3|_{P1}+...+|C1|_{PN}*|C2|_{PN}*|C3|_{PN}-Overlap_{P1,P2,...,PN}(C1xC2xC3)} \quad \textbf{(4.60)}$$

and the completeness formula (4.53) will become

$com_{P1,P2,...,PN}(r1 \bowtie r2)=com_{P1}(r1 \bowtie r2)+\ldots+com_{PN}(r1 \bowtie r2)-overlap_{P1,P2,...,PN}(r1 \bowtie r2)$ ***(4.61)***

Since we consider more than two peers, we assume only independence overlap among them. The changes in the formula for each overlap case are obvious and thus not presented. In addition, it is clear that the same formulae hold when we have a join of two or more properties with subsumed domain/range classes.

To generalize, let us assume that as in section 3.7, we have N peers which export the PDMS fragment F shown in Figure 13. Then, using the overlap estimations of Table 3.9 and formulae (4.59), (4.60) and (4.61) we can propose the following formulae for coverage, density and completeness estimations in the case of independence overlap.

$cov_{P1,P2,...,PN}(F)=cov_{P1}(F)+\ldots+cov_{P2}(F) + \ldots + cov_{PN}(F) - overlap_{P1,...,PN}(C1)*$
$overlap_{P1,...,PN}(C2)*\ldots* overlap_{P1,...,PN}(CN) = \Sigma_i cov_{Pi}(F) - \Pi_j overlap_{Pj}(C_j)$ ***(4.62)***

where i = 1,…,N-1 and j = 1,…,N.

$den_{P1,P2...,PN}(F)=$

$$\frac{|F|_{P1}+...+|F|_{PN}-Overlap_{P1,P2,...,PN}(F)}{|C1|_{P1}*...*|C_N|_{P1}+...+|C1|_{PN}*...*|C_N|_{PN}-Overlap_{P1,P2,...,PN}(C1x...xC_N)} \quad \textbf{(4.63)}$$

$com_{P1,P2,\ldots,PN}(F)=com_{P1}(F)+\ldots+com_{P2}(F) + \ldots + com_{PN}(F) - overlap_{P1,P2,\ldots,PN}(F) =$

$\Sigma_i com_{Pi}(F) - \Pi_j overlap_{Pj}(F)$ ***(4.64)***

We should state that formulae (4.62), (4.63) and (4.64) also hold for an arbitrary number of overlapping peers and an arbitrary fragment F, which involves joins over a set of classes.

4.6 DATA QUALITY OF QUERY PLANS

Let us consider peers P9, P14 and P15 of Figure 3. We will denote F1 = r3 ⋈ r1 and F2 = r2 ⋈ r6. As shown in Figure 3, peer P9 exports fragment F1, relating classes C1, C2 and C4. Let us assume that there is another peer P8 which also exports F1 with the same classes. Peers P14 and P15 export fragment F2, relating classes C2, C3 and C10. The set of peers that can answer fragment F1, will be denoted as SP1 = {P8, P9} and the set of peers that can answer F2 will be denoted as SP2 = {P14, P15}. Fragments F1 and F2 are joined on class C2. Then, for the query plan QP1 = F1 ⋈ F2 we will have

$$cov_{QP1}(F1_{SP1} \bowtie F2_{SP2}) = \frac{|C1|_{SP1} * |C2|_{SP1} * |C4|_{SP1} * |C2|_{SP2} * |C3|_{SP2} * |C10|_{SP2}}{||C2||} / ||C1|| *$$

$||C2|| * ||C3|| * ||C4|| * ||C10|| = cov_{SP1}(F1) * cov_{SP2}(F2)$ ***(4.68)***

$$den_{QP1}(F1_{SP1} \bowtie F2_{SP2}) = \frac{|F1 \times F2|_{QP1}}{|C1|_{SP1} * |C2|_{SP1} * |C4|_{SP1} * |C2|_{SP2} * |C3|_{SP2} * |C10|_{SP2} / ||C2||} \quad \textbf{(4.69)}$$

$$com_{QP1}(F1_{SP1} \bowtie F2_{SP2}) = \frac{|F1 \times F2|_{QP1}}{||C1|| * |C2|| * |C3|| * |C4|| * |C10||} = cov_{QP1}(F1_{SP1} \bowtie F2_{SP2}) * den_{QP1}(F1_{SP1} \bowtie F2_{SP2}) =$$

$$= \frac{|F1|_{SP1} * |F2|_{SP2} / ||C2||}{||C1|| * |C2|| * |C3|| * |C4|| * |C10||} = \frac{|F1|_{SP1}}{||C1|| * |C2|| * |C4||} * \frac{|F2|_{SP2}}{||C2|| * |C3|| * |C10||} =$$

$com_{SP1}(F1) * com_{SP2}(F2)$ ***(4.70)***

4.6.1 GENERALIZATION FOR MORE THAN TWO PEERS

Formulae (4.68), (4.69) and (4.70) can be applied also to fragments from an arbitrary number of peers, which involve an arbitrary number of properties that join pairwise on common classes. For example, let us assume that the set of peers SP1 export fragment F1, the set of peers SP2

export fragment F2,…, the set of peers SPn export fragment Pn. Then, the coverage and completeness formulae for the query plan QP = $F1_{SP1} \bowtie F2_{SP2} \bowtie \ldots \bowtie Fn_{SPn}$ will be

$$cov_{QP}(F1_{SP1} \bowtie F2_{SP2} \bowtie \ldots \bowtie Fn_{SPn}) = cov_{SP1}(F1) * cov_{SP2}(F2) * \ldots * cov_{SPn}(Fn) \quad \textbf{(4.71)}$$

$$com_{QP}(F1_{P1} \bowtie F2_{P2} \bowtie \ldots \bowtie Fn_{Pn}) = com_{SP1}(F1) * com_{SP2}(F2) * \ldots * com_{SPn}(Fn) \quad \textbf{(4.72)}$$

4.6.2 QUERY PLAN DATA QUALITY EXAMPLE

In favor of simplicity in the example, we assume that peers P9, P12, P13, P14 and P15 shown in Figure 3 are the only peers to export fragments in the PDMS. A peer that exports fragment F also exports the subfragments of F. As a result, it can answer questions concerning not only the fragment it exports, but also any of its subfragments (vertical subsumption). For example, peer P12 exports the fragment r1 ⋈ r2, and also its subfragments, i.e. properties r1 and r2. Thus, it can answer queries concerning not only r1 ⋈ r2, but also r1 or r2. Besides, a specific property of the query fragment can be answered not only by peers that export it, but also by peers that export a subproperty of it (horizontal subsumption). For example, peer P13 exports fragment r4 ⋈ r2 and r4 is a subproperty of r1, so P13 can also contribute to the answer of r1 ⋈ r2. Let us consider query F_{1236} and its possible fragmentations shown in Figure 4 as well as the peers of Table 2.3 answering each (sub)fragment illustrated in Figure 4.

Taking the fragment cardinalities of Table 3.1 into account, Table 4.15 shows the density of each peer with respect to the fragment it exports and its subfragments using formula (4.12). In this table we have also a column for fragment F_4, i.e. property r4, since it is a subproperty of r1 (fragment F_1) and thus it can contribute to the answer of fragments involving F_1.

Peers	F_{1236}	F_{123}	F_{126}	F_{12}	F_{13}	F_{26}	F_1	F_2	F_3	F_6	F_4	F_{24}
P9					0,5		0,5		1			
P12				1			1	1				
P13								0,5			1	0,5
P14	0,06	0,08	0,08	0,17	0,17	0,25	0,33	0,5	0,5	0,75		
P15						0,5		0,67		0,75		

Table 4.15 : Density information for the PDMS peers

Tables 4.16 and 4.17 show the coverage and completeness scores respectively for each peer using formulae (4.11) and (4.13) for the coverage and completeness of arbitrary fragments. These metrics are computed by the respective guide peers when needed.

Peers	F_{1236}	F_{123}	F_{126}	F_{12}	F_{13}	F_{26}	F_1	F_2	F_3	F_6	F_4	F_{24}
P9					0,2		0,4		0,33			
P12				0,03			0,07	0,1				
P13								0,4			0,13	0,13
P14	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	1	1		
P15						0,6		0,6		1		

Table 4.16 : Coverage information for the PDMS peers

Peers	F_{1236}	F_{123}	F_{126}	F_{12}	F_{13}	F_{26}	F_1	F_2	F_3	F_6	F_4	F_{42}
P9					0,1		0,2		0,33			
P12				0,03			0,07	0,1				
P13								0,2			0,07	0,07
P14	0,025	0,05	0,05	0,1	0,1	0,15	0,2	0,3	0,5	0,75		
P15						0,3		0,4		0,75		

Table 4.17 : Completeness information for the PDMS peers

In the following, we will consider plans that for each of their subfragments involve the union of peers that can answer it. In Figure 4 we can see that during the first round (0 joins) of the query routing and planning process we obtain the whole query fragment, F_{1236} which can be answered only by peer P14. According to Table 4.17 the completeness of this plan, let us call it QP1, is com(QP1) = 0,025. During the second round (1 join) we get the respective fragmentations of Figure 4. According to Table 2.3, for each (sub)fragment there may be more than one peer that can answer it. From the second round we get three plans, one for each fragmentation.

$QP2 = F_{123\{P14\}} \bowtie F_{6\{P14,P15\}}$

In order to compute the completeness of this plan, we should first compute $com_{P14,P15}(F_6)$. Using formula (4.23) it is

$com_{P14,P15}(F_6) = com_{P14}(F_6) + com_{P15}(F_6) - com_{P14}(F_6) * com_{P15}(F_6) = 0,75 + 0,75 - 0,75*0,75 = 1,5 - 0,56 = 0,94$

Then, taking formula (4.70) into account it is

$com(QP2) = com_{P14}(F_{123}) * com_{P14,P15}(F_6) = 0,05 * 0,94 = 0,047$

Taking Table 2.3 into account, the plan that occurs from fragmentation $F_{126} \bowtie F_3$ is

$QP3 = F_{126\{P14\}} \bowtie F_{3\{P9,P14\}}$

In order to compute the completeness of this plan, we should first compute $com_{P9,P14}(F_3)$. Using formula (4.23) it is

$com_{P9,P14}(F_3) = com_{P9}(F_3) + com_{P14}(F_3) - com_{P9}(F_3) * com_{P14}(F_3) = 0,33 + 0,5 - 0,33 * 0,5 = 0,83 - 0,17 = 0,66$

Then, using formula (4.70) it is

$com(QP3) = com_{P14}(F_{126}) * com_{P9,P14}(F_3) = 0,05 * 0,66 = 0,033$

Considering Table 2.3, the plan that occurs from fragmentation $F_{13} \bowtie F_{26}$ is

$QP4 = F_{13\{P9,P14\}} \bowtie F_{26\{P14,P15\}}$

In order to compute the completeness of this plan, we should first compute $com_{P9,P14}(F_{13})$ and $com_{P14,P15}(F_{26})$. Taking formula (4.23) into account it is

$com_{P9,P14}(F_{13}) = com_{P9}(F_{13}) + com_{P14}(F_{13}) - com_{P9}(F_{13}) * com_{P14}(F_{13}) = 0,1 + 0,1 - 0,1 * 0,1 = 0,2 - 0,01 = 0,19$

$com_{P14,P15}(F_{26}) = com_{P14}(F_{26}) + com_{P15}(F_{26}) - com_{P14}(F_{26}) * com_{P15}(F_{26}) = 0,15 + 0,3 - 0,15 * 0,3 = 0,45 - 0,05 = 0,4$

Then, using formula (4.70) it is

$com(QP4) = com_{P9,P14}(F_{13}) * com_{P14,P15}(F_{26}) = 0,19 * 0,4 = 0,076$

During the third round (2 joins) we get the respective fragmentations of Figure 4. From the third round we get three plans, one for each fragmentation.

$QP5 = F_{12\{P12,P13,P14\}} \bowtie F_{3\{P9,P14\}} \bowtie F_{6\{P14,P15\}}$

$QP6 = F_{26\{P14,P15\}} \bowtie F_{1\{P9,P12,P13,P14\}} \bowtie F_{3\{P9,P14\}}$

$QP7 = F_{13\{P9,P14\}} \bowtie F_{2\{P12,P13,P14,P15\}} \bowtie F_{6\{P14,P15\}}$

In order to compute the completeness of each plan, we should first compute the completeness for each of its subfragments as above. We should point out that peer P13 exports fragment F_{42}, where F_4 is a subproperty of F_1 and thus it can also contribute to plans concerning F_{12} and F_1. It is

$com_{P12,P13,P14}(F_{12}) = com_{P12}(F_{12}) + com_{P13}(F_{42}) + com_{P14}(F_{12}) - com_{P12}(F_{12}) * com_{P13}(F_{42}) * com_{P14}(F_{12}) \approx 0,03 + 0,01 + 0,1 \approx 0,14$

After the computations we have

$com(QP5) = com_{P12,P13,P14}(F_{12}) * com_{P9,P14}(F_3) * com_{P14,P15}(F_6) = 0,14 * 0,66 * 0,94 = 0,087$

$com(QP6) = com_{P14,P15}(F_{26}) * com_{P9,P12,P13,P14}(F_1) * com_{P9,P14}(F_3) \approx 0,4 * 0,54 * 0,66 = 0,143$

$com(QP7) = com_{P9,P14}(F_{13}) * com_{P12,P13,P14,P15}(F_2) * com_{P14,P15}(F_6) \approx 0,19 * 1 * 0,94 = 0,179$

During the fourth round (3 joins) we get only one fragmentation in Figure 4 and thus only one plan.

$QP8 = F_{1\{P9,P12,P13,P14\}} \bowtie F_{2\{P12,P13,P14,P15\}} \bowtie F_{3\{P9,P14\}} \bowtie F_{6\{P14,P15\}}$

The completeness of QP8 is computed as follows:

$com(QP8) = com_{P9,P12,P13,P14}(F_1) * com_{P12,P13,P14,P15}(F_2) * com_{P9,P14}(F_3) * com_{P14,P15}(F_6) \approx 0,54 * 1 * 0,66 * 0,94 \approx 0,335$

The completeness of the eight plans is shown in Table 4.18 :

Plans	**Completeness**	**Joins in each plan**
QP1	0,025	F_{1236}
QP2	0,047	$F_{123\{P14\}} \bowtie F_{6\{P14,P15\}}$
QP3	0,033	$F_{126\{P14\}} \bowtie F_{3\{P9,P14\}}$
QP4	0,076	$F_{13\{P9,P14\}} \bowtie F_{26\{P14,P15\}}$
QP5	0,087	$F_{12\{P12,P13,P14\}} \bowtie F_{3\{P9,P14\}} \bowtie F_{6\{P14,P15\}}$
QP6	0,143	$F_{26\{P14,P15\}} \bowtie F_{1\{P9,P12,P13,P14\}} \bowtie F_{3\{P9,P14\}}$
QP7	0,179	$F_{13\{P9,P14\}} \bowtie F_{2\{P12,P13,P14,P15\}} \bowtie F_{6\{P14,P15\}}$
QP8	0,335	$F_{1\{P9,P12,P13,P14\}} \bowtie F_{2\{P12,P13,P14,P15\}} \bowtie F_{3\{P9,P14\}} \bowtie F_{6\{P14,P15\}}$

Table 4.18 : Completeness of plans QP1 – QP8

It is clear that among the eight plans of Figure 4, plan QP8 has the highest completeness score. This was expected, since QP8 has the maximum number of joins and thus involves the most answers to the query fragment. None of these query plans has value 1, as according to

formula (4.13) that we use to compute completeness, the denominator, which represents the number of possible fragment instances in the PDMS (i.e. the size of the Cartesian product of all the instances of the classes that compose the fragment in the PDMS), is always greater than the numerator, i.e. the actual number of fragment instances.

The advantage of this formula is that it is easier to express composed completeness. Also, it can be used as a measure for completeness when comparing plans (it is monotonic), but the drawback is that if we want to evaluate the "closeness" to the complete answer set, we must evaluate the completeness value for the largest plan and compare the value of the other plans to it. For example, let us take the completeness of QP7. This value has no meaning alone, but only compared to the maximum completeness, that of QP8.

4.7 VALUE RANGE OF DATA QUALITY METRICS

In the previous sections we presented definitions for the coverage, density and completeness data quality metrics and we proposed formulae for data quality estimation for simple or complex fragments, in different overlap cases and for one or more peers. In this section we are going to examine the range of the values of the data quality formulae discussed already.

Let us first consider the case of one peer exporting a fragment F, either simple, (class or property) or complex (comprising two or more properties). Coverage formula (4.1), density formula (4.2) and completeness formula (4.3) give always values in [0, 1], since the denominator is always greater than the numerator. However, when it comes to the union of peers with respect to a fragment, some differences occur.

The simplest case is that of the union of two or more peers with respect to a single class. Then only completeness of the union is considered. The completeness value of formula (4.14) can be greater than 1. Its greatest value occurs when the peers are disjoint with respect to the exported class.

Formula (4.16) shows the case of peers exporting subsumed classes. The greatest value for this formula occurs when the class (e.g. C1) in one peer and its subclass (e.g. C5) in the other peer are disjoint. However, due to the existence of subsumption between the classes, our overlap

estimations may be smaller than reality and thus the completeness of the union may be greater than 1.

When two or more peers export the same property, either with the same domain/range or with subsumed domains/ranges, coverage values, as we can see in formula (4.19) are usually in [0, 1], however, for the same reasons as previously, they may be greater than 1, especially in the worst case, when the peers are disjoint with respect to the specific property. Then, no overlap of the domain and range classes of the property is involved in the numerator and the sum of the maximum possible instances of the property in the two peers may be greater than the product of the classes that the property comprises. In addition, it is clear that the density formula (4.21) always gives values in [0, 1], since the numerator involves the property instances in the two peers and the overlap of these peers with respect to this property, while the denominator involves the maximum possible property instances in these peers and their overlap with respect to the classes that form the property. Thus, the fraction is always ≤ 1. As for completeness of the union of two or more peers with respect to a property, the score is most of the times in [0,1], as shown in formula (4.22), since the numerator involves the property instances of the peers, while the denominator involves the maximum possible instances of these peers with respect to this property. However, if a great percentage of the class instances are connected to form properties or more complex fragments, the value of completeness may be > 1, since it is a sum and not a product of completeness values. These conclusions also hold for the union of peers with respect to subsumed properties and with respect to more complex fragments. We should point out that in our estimations we will use 1 as upper bound value for coverage and completeness scores, when they happen to be greater than 1.

As far as query plans are concerned, formulae (4.71) and (4.72) show that the coverage and completeness of a query plan involving arbitrary fragments among an arbitrary number of peers give always values in [0, 1]. This is easy to understand, since the coverage of a query plan is formed as the product of the coverage of every fragment it comprises. The coverage value are in [0, 1] and thus their product is also in [0, 1]. The same holds for the completeness formula, which is formed as the product of the completeness values of every fragment the query plan comprises.

In this section we discussed the value range for data quality metrics in the case of one peer and in the case of two or more peers that export the same or subsumed fragments.

4.8 RELATED WORK

In this section we compare our data quality metrics with respect to similar metrics in the literature.

The closer work to our framework is **[4]**. Before detailing the introduced data quality metrics, we explain the differences between the universal relational model and RDF/s. In **[4]**, we say that a source contains a specific tuple of the universal relation, even if the source does not provide values for all the attributes in the tuple. However, in the RDF/S data model that we use in our work, a peer P may contain instances of classes appearing as domain or range of properties, but this does not imply that these class instances are actually related thought properties (i.e. optional property instances). As a result, we need to distinguish in our RDF/S-based PDMS between the maximum expected number of p1 instances in peer P, denoted as $|C1|_P * |C2|_P$, and the exact number of p1 instances, denoted as $|p1|_P$. Apart from this difference, we must express every data quality metric of a peer P with respect to a specific fragment, (i.e. a class, a property or a more complex fragment) exported by peer P, while in **[4]** data quality metrics are defined for a source.

In **[4]**, for every source, coverage of a source is defined as the ratio of the number of tuples of the universal relation the source stores, to the size of the universal relation. Coverage scores are in [0, 1]. In contrast, we define the coverage of a peer P with respect to a fragment F as the ratio of the maximum number of F instances peer P can export, to the total maximum number of F instances exported in the whole PDMS. For example if F is a property p1 with domain class C1 and range class C2, the maximum number of p1 instances is $|C1|_P * |C2|_P$ in that peer. In case F is a complex fragment consisting of k properties p1, p2, …, pk, involving classes C1, C2, …, Ck+1 we consider the product $|C1|_P*|C2|_P*\ldots*|Ck+1|_P$ as the maximum number of F instances in peer P.

There are three definitions of density of a source S in **[4]**, one called *attribute* density, with respect to a certain attribute of the relation the source exports, one called *source* density, with respect to all the attributes of the universal relation and one called *query – dependent* density. Attribute density with respect to attribute a, is defined as the ratio of the tuples of the source S that have a non-null value for a, to the total number of tuples in S. Source density of source S is the average density of S over all attributes of the universal relation. Query - dependent density of a source S with respect to a query Q is the average density of S over all attributes of Q. It is obvious that the scores of the three types of density are in [0, 1].

In our RDF/S model, properties of a fragment in a peer play the role of attributes of a relation in a source. When class instances in a peer are connected together to form property instances, or properties are joined to form more complex fragments, then the specific property has instances in the fragment of this peer, in the same way that an attribute may have a non-null value in the relation exported by a source. Thus, density of a peer P with respect to a fragment F is a local estimation of the actual percentage of F instances we could obtain from peer P w.r.t. the maximum number of expected F instances in this peer base. It is clear that our density scores are in [0, 1]. Our definition of density corresponds to the source density notion of **[4]**, but given that a peer exports also all of its sub-fragments it also captures the notion of attribute density for RDF/S.

As far as completeness of a source S is concerned, in **[4]** it is defined as the ratio of the number of tuples in S for which all attributes of it have non – null values, to the number of tuples of the completely filled universal relation. This is the product of the number of tuples in the universal relation and the number of attributes of the universal relation. The completeness of a source is the product of its coverage and density. Completeness scores are in [0, 1].

In our approach we define completeness of a peer P with respect to a fragment F as the probability that a random F instance of the PDMS appears in peer P. Completeness of peer P with respect to a fragment F is the product of coverage and density of P with respect to F. Since the coverage and density values in our case are always smaller than 1, the completeness scores are always smaller than 1. Our completeness formula is similar to the coverage formula presented in **[4]**. Another important remark is that in **[4]** density is defined not only for a single source, but also for more sources. Likewise, our definition of density can be applied to either the union or join of two or more peers exporting either simple or complex fragments.

The mediator-based system presented in **[28]**, also relies on the relational data model. Every source can provide objects (tuples) each of which has a globally unique object identifier and a set of attributes. Attributes of an object can be stored in different sources, as long as each source retains the identifier of the specific object. A source represents a set of objects. The global schema of the integration information system consists of only one global relation, which contains all object identifiers and the union of all attributes delivered by sources. The closed – world assumption holds as to the attributes that the sources export. In order to define density of merged results between two sources, the notion of full – outerjoin merge is used : the final result will

involve the tuples that appear only in each one of the sources and their common tuples. Using the RDF data model in our framework, we have a notion similar to the full – outerjoin merge with the existence of sub-fragments of a fragment F in a peer. Their cardinalitites are greater than that of F, since they involve both instances that are connected to form F instances and instances that are not connected.

A density measure is used to guide planning in this framework and provide the user with the best plans with respect to this metric. Density in this work is perceived as the "fullness" of the sources, i.e. the degree to which they provide non – null attributes for the objects they contain. Two definitions of density are presented, density of an attribute of a source S and density of a source S with respect to a query Q. The definitions are the same as those presented in **[4]**, with values in [0, 1]. Besides, density of a source is also defined with respect to weightings provided by the user for the attributes of a user query. In this case, density is calculated as the weighted average of attributes involved in the user query, with values in [0, 1].

In the case of merging the results of two queries in different sources into a single result, three parts are generated: the first part consists of objects contained only in the first source, the second part consists of objects covered by both sources and the third part consists of objects covered only by the second source. The second part is the result of a join over the two results using the object identifier as the join predicate. Objects with the same identifier are merged into one. The attributes of a merged result may be attributes covered by only one of the sources, attributes covered by both sources, or attributes that are not covered at all. If an attribute is covered by only one of the sources, then the values of this attribute in the result will be the ones contained in this source. If none of the sources provide values for some attributes (i.e. these attributes have null values in both sources), then the specific attributes will have null values in the result. If both sources provide non – null values for a specific attribute, then a resolution function will determine what value will appear in the result. Formulae for density scores of an attribute whose values are merged between two sources are provided, for different overlap cases (i.e. disjointness, independence, containment of the objects provided by one source to the other, equality of the objects the two sources provide attributes for). We do not further discuss these formulae, as they are presented in **[4]**.

In **[6]**, query planning for relational mediators is presented. Various quality criteria are presented, originally defined in **[34]** and each source and query plan achieves scores in each

criterion. The assumption of independence among the criteria is made. The scores of a source or a plan in all criteria are then aggregated to form a single overall information quality score that characterizes it. Then, sources and plans are ranked with respect to their aggregated information quality scores. Some of the quality criteria addressed in this work are reliability and timeliness of a source, relevancy, and availability of sources. The only similarity to our data quality metrics is the notion of completeness of a source relation, which is defined as the sum over the percentages of non-null values in each attribute of the relation, adjusted by a user weighting, stating the importance of each attribute. However, even though quality scores are assigned to sources and plans, no formulae are proposed for any of these criteria.

In **[29]**, the notion of coverage is used as a quality metric of web sources represented using the relational data model. In this framework, the size of a source is the number of objects it contains. The universe is the relation that contains all possible objects. The coverage of a source is defined as the ratio between the size of the source and the size of the universe. The coverage of a mediated query is the total number of distinct objects the query is computed on, divided by the size of the universe. This means that coverage scores in both cases are in [0, 1]. For the overlap cases defined in this work, the following formulae are proposed for the coverage of two sources R and S with respect to a mediated query P:

a) If R and S are disjoint, coverage(P) = coverage(R) + coverage (S)

b) If R and S are equivalent, coverage(P) = coverage(R) = coverage(S)

c) If R is a subset of S, coverage(P) = coverage(S)

d) If R and S are independent, coverage(P) = coverage(R) + coverage(S) – coverage(R) * coverage(S)

The disjointness case is similar to ours : Since there is no overlap between the sources, the total coverage is the sum of the coverage of each source. As for the other cases, there is a main difference between this approach and our approach: We consider overlap with respect to the class instances of the peers, while they consider overlap with respect to the relation the peers export. In addition, their definition of coverage is similar to our definition of completeness. As a result, in the fourth overlap case the coverage of the plan P is the sum of the coverage of R and S if we subtract their product, which represents independence overlap. For the case of more than two sources, the combined coverage of the first two sources is computed and then the sources are

treated as a single source, combined with the third sources and so on. However, there is no statement that the coverage scores for two or more peers are always ≤ 1.

Chapter 5
CONCLUSIONS AND FUTURE WORK

P2P data management systems (PDMSs), are capable of supporting loosely coupled communities of databases in which each peer base can join and leave the network at free will, while groups of peers can collaborate on the fly to provide advanced data management services on a very large scale (i.e., thousands of peers, massive data).

However, as the number of peers in a PDMS increases and queries become complex (e.g. tree or graph-shaped), the number of produced plans that need to be optimized and executed becomes huge. As a result, there is a need to prune the search space by considering quality metrics of the data hosted by each peer. We should be able to rank the peers contributing to a plan, and thus the plans themselves, according to data quality metrics allowing to discard plans producing poor quality query results (according to a threshold either set by the user or the system).

In this thesis, we considered data quality metrics of the view instances published by the peers with respect to the PDMS schema and its virtual instantiation. In particular, we provided definitions and formulae for the *coverage*, *density* and *completeness* data quality metrics in order to estimate the data quality of peer databases and query plans according to the RDFS schema fragments they involve, either simple or complex. We also addressed the notion of *overlap* between peers with respect to a schema fragment and proposed formulae for its estimation.

Our aim for future work is to present a metric that combines cost and completeness of plans and enrich existing query planning algorithms proposed for RDF-based PDMSs, in order to prune the huge search space of plans that is created during query planning. Our objective is to discard plans that are ranked below a specific data-quality threshold and thus reduce as much as possible the planning time, while ensure that the final plan to be executed will be the best possible one with respect to the enforced data quality constraints. An interesting idea in this context is a query optimization process that interleaves query routing and planning enriched with the data quality metrics we provide as well as traditional distributed cost optimization heuristics.

In addition, our goal is to experimentally illustrate the gains of our approach for pruning the search space and thus improving the overall query processing time and quality with respect to

PDMSs of increasing size and queries of increasing complexity. We intend to use the SQPeer simulator proposed in **[1]** for our experiments, which is based on the ubQL^{VM} proposed in **[9]**. The SQPeer simulator at the time being uses a specific cost model presented in **[1]** and the Dynamic and Iterative Dynamic Programming algorithms for query planning. The next step is to run experiments on the simulator using our combined cost and completeness metric, to observe the benefits of pruning the space of plans both with respect to cost and quality.

In addition, we considered upper bound values for some data quality metrics we defined, when their estimations were greater than 1. As future work we could provide better estimation values for some of our formulae, thus no upper bound values would be necessary.

Another interesting target would be to provide definitions and estimation formulae in a PDMS context for more quality metrics, as the ones presented in **[6]** and **[20]**. A great challenge is the multi-objective query optimization, which considers not only the data quality characteristics that we address, but also others, such as availability of a data source (the percentage of time that the source is accessible), timeliness (update-frequency of the data in the source), etc. Most of such characteristics are considered to be highly subjective, so objective metrics need to be defined for them. We could then achieve an even better pruning of the search space in query planning and return plans of even higher quality to the user.

BIBLIOGRAPHY

[1] Semantic Query Routing and Planning in Peer-to-Peer Database Systems, The SQPeer Middleware, G. Kokkinidis, Master's Thesis, University of Crete, 2005.

[2] Indexing Views to Route and Plan Queries in a PDMS. L. Sidirourgos, Master's thesis, University of Crete, Computer Science Department, 2005.

[3] Query Processing in RDF/S – based P2P Database Systems, G. Kokkinidis, L. Sidirourgos, V. Christophides, at Semantic Web and Peer – to – Peer, S. Staab, H. Stuckenschmidt (eds.), Springer – Verlag, 2005.

[4] Completeness of Integrated Information Sources, F. Naumann, J. C. Freytag, U. Leser, Technical Report HUB-IB-135, February 2000.

[5] Using probabilistic information in data integration, D. Florescu, D. Koller, and A. Levy, In Proceedings of the International Conference on Very Large Data Bases (VLDB), 1997.

[6] Quality driven Integration of Heterogeneous Information Systems, F. Naumann, J. C. Freytag, U. Leser, In Proceedings of the International Conference on Very Large Data Bases (VLDB), Edinburgh, 1999.

[7] Mining Source Coverage Statistics for Data Integration, Z. Nie, S. Kambhampati, U. Nambiar, S. Vaddi, Proceedings of Workshop on Web Information and Data Management, 2001.

[8] A Frequency – based Approach for Mining Coverage Statistics in Data Integration, Z. Nie, S. Kambhampati, In Proceedings of the ICDE Conference, 2004.

[9] ubQL: A Distributed Query Language to Program Distributed Query Systems, A. Sahuguet, PhD thesis, University of Pennsylvania, 2002.

[10] Viewing the Semantic Web Through RVL Lenses., A. Magkanaraki, V. Tannen, V. Christophides, and D. Plexousakis, In Proceedings of the 2nd International Semantic Web Conference (ISWC), 2003.

[11] Semantic Overlay Networks for P2P Systems, Technical report, A. Crespo and H. Garcia-Molina, Computer Science Department, Stanford University, 2003.

[12] Towards High Performance Peer-to-Peer Content and Resource Sharing Systems, P. Triantafillou., C. Xiruhaki, M. Koubarakis, and N. Ntarmos, In Proceedings of the International Conference on Innovative Data Systems Research (CIDR), January 2003.

[13] Data Management for Peer-to-Peer Computing: A Vision., P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. In Proceedings of the 5th International Workshop on the Web and Databases (WebDB), Madison, Wisconsin, 2002.

[14] Piazza: Data Management Infrastructure for Semantic Web Applications., A. Halevy, Z. Ives, P. Mork, and I. Tatarinov. In Proceedings of the 12th Conference on World Wide Web (WWW), Budapest, Hungary, 2003.

[15] Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks., W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Loser., In Proceedings of the 12th International Conference on World Wide Web (WWW), Budapest, Hungary, 2003.

[16] Resource Description Framework (RDF). http://www.w3.org/RDF/

[17] Benchmarking RDF Schemas for the Semantic Web., A. Magkanaraki, S. Alexaki, V. Christophides, and D. Plexousakis, In Proceedings of the 1st International Semantic Web Conference (ISWC'02), 2002.

[18] RQL: A Declarative Query Language for RDF, G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl, In Proceedings of the 11th International Conference on World Wide Web (WWW), Honolulu, Hawaii, USA, 2002.

[19] Database Management Systems, R. Ramakrishnan, J. Gehrke, 2nd Edition, McGraw-Hill, 2000

[20] Data quality assessment from the user's perspective, C. Cappiello, C. Francalanci, B. Pernici, In Proceedings of the International Workshop on Information Quality in Information Systems, Paris, France, 2004.

[21] Efficiently Ordering Query Plans for Data Integration, A. Doan and A.Levy, In Proceedings of the 18th International Conference on Data Engineering ICDE, 2002.

[22] Assessment Methods for Information Quality Criteria, F. Naumann, C. Rolker, In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 2000.

[23] Joint Optimization of Cost and Coverage of Information Gatherinh Plans, Z. Nie and S. Kambhampati, In Proceedings of the 10th International Conference on Information and Knowledge Management CIKM, 2001.

[24] Cardinality Estimation for the Optimization of Queries on Ontologies, E.P. Shironoshita, M. T. Ryan, M.R. Kabuka, SIGMOD Rec. 36, 2, 13 – 18, 2007

[25] The State of the Art in Distributed Query Processing, D. Kossmann, ACM Computing Surveys 32(4), pp. 422-469, December 2000

[26] Iterative Dynamic Programming : A New Class of Query Optimization Algorithms, D. Kossmann, K. Stocker, ACM Transactions on Database Systems, 25(1), March 2000

[27] Havasu : A multi-objective, adaptive query processing framework for data integration, S. Kambhampati, U. Nambiar, Z. Nie and S. Vaddi, ASU CSE Technical Report - 02-2005

[28] Density Scores for Cooperative Query Answering, F. Naumann and U. Leser, In Proc. of the Workshop Föderierte Datenbanken, p. 103–116, Berlin, 1999.

[29] Maximizing Coverage of Mediated Web Queries, R. Yerneni, F. Naumann, and H. Garcia-Molina, Technical Report, Stanford University, 2000.

[30] Mining coverage statistics for websource selection in a mediator, Z. Nie, U. Nambiar, S. Vaddi, and S. Kambhampati.. ASU Technical Report, 2002.

[31] Querying heterogeneous information sources using source descriptions, A. Y. Levy, A. Rajaraman, J. J. Ordille, In 22th VLDB, p. 251 - 262, Bombay, India, 1996

[32] Efficient decision theoretic planning: Techniques and empirical analysis, P. Haddawy, A. Doan, and R. Goodwin, In Proc. of the Nat. Conf. on Uncertainty in AI (UAI), 1995.

[33] Access path selection in a relational database management system, P. Selinger, M. Astrahan, D. Chamberlin, R. Lorie, T. Price, In SIGMOD ’79, 1979.

[34] Beyond accuracy: What data quality means to data consumers. Richard Y. Wang and Diane M. Strong, Journal on Management of Information Systems, 12, 4:5-34, 1996.

[35] Principles of Distributed Database Systems, M.T. Ozsu and P. Valduriez. Prentice Hall, 1991.