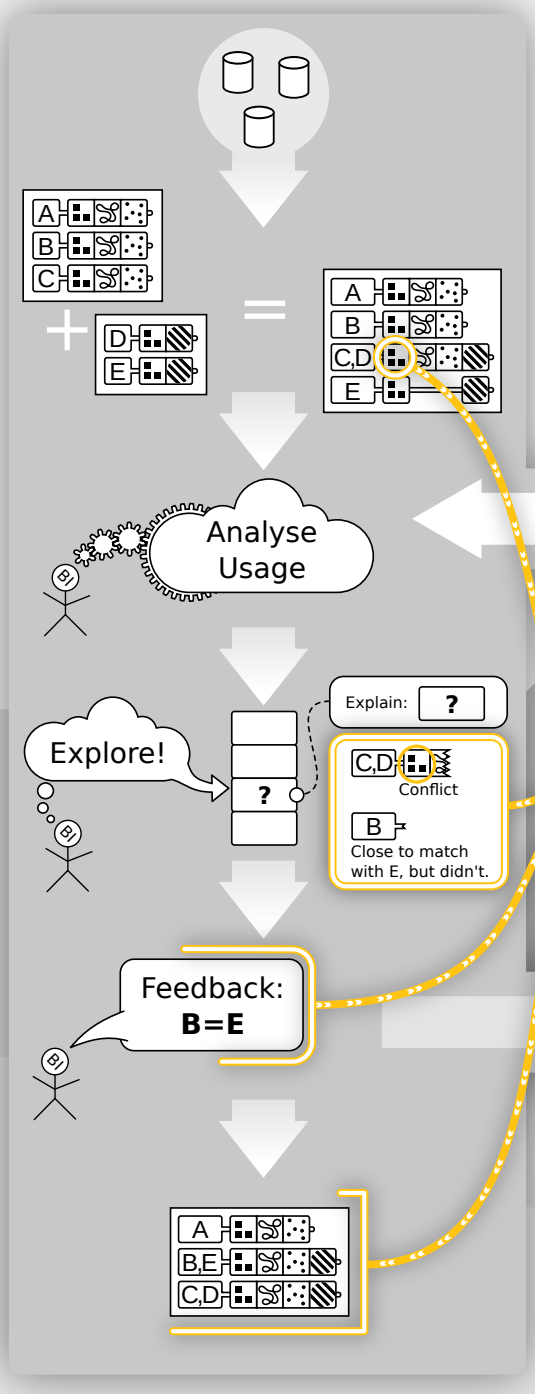
Repurposing and Probabilistic Integration of Data

An iterative and data model independent approach

Brend Wanders



Repurposing and Probabilistic Integration of Data

An iterative and data model independent approach

Brend Wanders

Graduation committee:

Chairman:	Prof. dr. Peter M.G. Apers
Promoter:	Prof. dr. Peter M.G. Apers
Assistant promoter:	Dr. ir. Maurice van Keulen

Members:

Prof. dr. Willem Jonker	University of Twente
Prof. dr. Jaco C. van de Pol	University of Twente
Prof. dr. Birgitta König-Ries	Friedrich-Schiller-Universität Jena
Prof. dr. Dan Olteanu	University of Oxford

CTIT Ph.D.-thesis Series No. 16-388
Centre for Telematics and Information Technology
University of Twente
P.O. Box 217, NL – 7500 AE Enschede

SIKS Dissertation Series No. 2016-24
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN: 978-90-365-4110-7
ISSN: 1380-3617 (CTIT Ph.D.-thesis Series No. 16-388)
DOI: 10.3990/1.9789036541107
Available online at `http://dx.doi.org/10.3990/1.9789036541107`

Cover design by Brend Wanders
Printed by Gildeprint

REPURPOSING AND PROBABILISTIC INTEGRATION OF DATA

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
Prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties,
in het openbaar te verdedigen
op donderdag 16 juni 2016 om 16.45 uur

door

Brend Wanders

geboren op 13 april 1985
te 's-Gravenhage

Dit proefschrift is goedgekeurd door:

Prof. dr. Peter. M.G. Apers (promotor)
Dr. ir. Maurice van Keulen (assistent-promotor)

"A twentieth century problem is that technology has become too "easy". When it was hard to do anything whether good or bad, enough time was taken so that the result was usually good. Now we can make things almost trivially, especially in software, but most of the designs are trivial as well. This is inverse vandalism: the making of things because you can. Couple this to even less sophisticated buyers and you have generated an exploitation marketplace similar to that set up for teenagers. A counter to this is to generate enormous dissatisfaction with one's designs using the entire history of human art as a standard and goal. Then the trick is to decouple the dissatisfaction from self worth — otherwise it is either too depressing or one stops too soon with trivial results."

— The Early History Of Smalltalk, Alan C. Kay

Preface

In the warm fall of 2011 I was finishing up my master's thesis at a leisurely pace. At some point during this time my supervisor, Paul van der Vet, surprised me by asking if I had interest in pursuing a Ph.D. with the database group.

Over the course of my education at the university I had come in contact with this thing called "research". My idea of what "research" actually entailed was almost completely shaped by the few courses that hoped to emulate academic research, and those succeeded only in the most mechanical manner. The prospect of drudging through four years of what I had come to see as "academic research" did not appeal to me.

During my studies I investigated, with much enthusiasm, a way to combine online text-based virtual worlds with a interactive narrative generator. For my master's thesis I worked together with very smart people and wrote code that allowed biochemists to explore the complex results from signalling pathway simulations. My contributions mattered, and real biochemists were happy to use what I wrote. However, I did not view these projects as "research", they lacked the mechanical and repetitive nature of "research" as I knew it.

I was at a crossroads, and did not know which way to go. I thought about the offer, I discussed the idea of doing research with people whose opinion I valued greatly, and then I thought about it some more. In the end my perspective on what it meant to do research shifted and I accepted the offer. So, with a renewed sense of urgency I soon finished my master's thesis, and started my new job as "assistent in opleiding".

Brend Wanders
Enschede, May 2016

Acknowledgements

First and foremost I would like to thank Maurice van Keulen and Paul van der Vet for their unwavering support during the creation of this book. Their experience with all things academic, both scientific and organisational, and the many insightful discussions about all manners of topics have been most helpful.

I would like to thank Niels Bloom and Ivor Wanders for their many comments on the draft of this thesis, and for their insights and work-arounds that have been of great value while working with LaTeX and editting in general.

Finally, I would like to thank my colleagues, friends and family for their continuous support.

Contents

CHAPTER 1

Introduction

Imagine you have been tasked with researching a hospital's diagnostic and treatment processes associated with pregnancy. This research has to be based on electronic patient dossiers (EPDs). The hospital would like to know what paths of consults and treatments their patients go through, to improve care and cut down on costs.

You know that the EPDs store a record of all consults and treatments for a patient. Obviously you need to extract those consult and treatment records that pertain to the pregnancies of the selected population of women.

You go through the motions of obtaining permission from the Ethics board to use anonymised data and obtaining access to the actual data. After a short e-mail conversation with your contact at the hospital, in which you ask about the encoding of the data files, you start by extracting all consults and treatment records.

You quickly discover many records not related to pregnancies after obtaining the first results from your analysis. Your assumption that all records of a pregnant woman during the pregnancy are related to the pregnancy is wrong: she may for example be treated for a condition she already had. There is, however, no objective means such as a field in the data that says 'related to pregnancy'.

So you embark on long and painstaking process where you define filter rules. You read up on the treatments and consultation types related to pregnancy and adjust the filters. Some records are easy, since the hospital ward they reference is dedicated to pregnancies. Other records are hard, since the diagnostic methods

referenced in them are used by many medical disciplines, and the equipment is shared amongst several departments of the hospital.

All the while you have to re-run your filtering on the original data and inspect sampled results by manually browsing through the results and spotting errors as you go. You do this knowing that you cannot catch all of the records. Some noisy records will remain.

You spend weeks fiddling with the filters and inspecting so many result rows that you now know the abbreviations for all diagnostic methods by heart. Finally, you are confident that enough of the noisy records have been filtered out. You can now start on the next step: determine the possible sequences of the consults and treatments.

Then, when looking at a sample, you notice something strange in the timestamps of consults: for a certain clinician many consults appear close to each other and in the evening.

You investigate this strange occurrence. After carefully looking at the consults for this clinician and comparing it with other consults that your filtering rules produce, you start to understand that the modification time of an EPD — which is the only time that is recorded — does not reflect the actual moment of the activity. Actually, after a closer look at the data, you are not even sure that the times you see reflect the order in which the activities took place during a day.

Your contact at the hospital tells you that the modification time of the EPD really is all the available data on the timing of a consult or treatment. You e-mail back that without good time data, the quality of this data set is too low for the purpose of this research. They follow up with "You are welcome to visit and see what's happening."

You decide to accept the invitation. A week later you are at the hospital, holding a cup of coffee and a notepad. Over the course of days you follow several clinicians around, scribbling notes in your notepad. You ask questions and track the work of two clinicians dealing with pregnancies.

After the first few days you see a pattern emerge: a clinician typically sees

many patients on one day and it is often too disruptive for him to update the EPD immediately. Hence, it is common practice that he updates the dossiers at the end of day or even later. Not necessarily in the order that he saw the patients.

This story is about a scientist attempting to reuse existing data for a new purpose. The original data is not collected for the analysis of a hospital's processes. So, the scientist struggles with data quality problems such as noisy records about treatments and consults not related to pregnancy.

The scientist struggles with semantics issues like the modification time of the EPD, which leads to a data quality issue about ordering the consults and treatments. Even on the practical side, the scientist struggles with defining filters and painstakingly has to investigating the results through manual inspection. In short, the scientist struggles to reuse and repurpose the data.

This thesis is about that struggle.

1 Global aims of this thesis We want to assist the process of repurposing data by developing generic technology assisting the process of data understanding and data combination.

Every scientist has their own way of working, and uses tools in their own way. To best assist the scientist, automated assistance should not enforce a specific pattern of work. Instead, such tools should work within the established workflow of the scientist.

We aim to support rapid feedback in the developed technologies we developed. Rapid feedback leads to faster understanding and refinement, which in turn leads to faster research.

1.1 Motivation

Jim Gray introduced the term "the fourth paradigm" to signify a revolution in scientific method [51]. Besides the paradigms of empiricism, mathematical

modelling, and simulation, the method of combining and analysing data in novel ways has become a main research paradigm capable of tackling research questions that could not be answered before.

2 Data intensive research New disciplines have emerged that separate data producers and consumers. For example, in physics and astronomy one group of researchers design, build and operate complex measurement equipment to gather data, while another group studies that data to determine and understand the laws that govern particles, celestial bodies, and other phenomena.

Another prominent example is bioinformatics which is "the development and use of computational methods for data management and data analysis of sequence data, protein structure determination, homology-based function prediction, and phylogeny." [54]

In many other disciplines, similar developments can be observed where researchers use data-driven methods to study phenomena based on available data. The social sciences have started to discover data analysis as a means to study human and crowd behaviour from various kinds of traces of human activity (e.g., [3, 43]).

Further examples are the analysis and reuse of content and structure from WikiPedia (e.g. [73, 9, 52]), recording and analysing traffic patterns in civil engineering, analysing software version management repositories for understanding collaboration patterns, etc. This data and analysis driven scientific method is often called e-science.

3 Collection of data All of these data driven disciplines have one thing in common: they need data sources. Many data sources are created specifically for research. Data for such sources is collected with a certain purpose in mind. The intended purpose of the data imposes certain requirements on the design of the organisation of the data.

The creation of research data sets is a slow, and often expensive, endeavour. To keep costs down and to get results faster, data sets are made with a strong

focus on their specific purpose. All collected data is organised in a way that facilitates the purpose of the data set, and to get research results more quickly.

In some cases the purpose, or part of the purpose, of a data set might be to share the collected data set with other researchers so they can use it in their research. Even if the data set is created with the express purpose of sharing the collected data, it will lend itself better to some uses than to others — the purpose of the data and the organisation of the data set influence each other.

4 Reuse of data: a struggle Combining and analysing data in novel ways is the reuse of data. Data reuse means taking an existing source of data and using it for a new purpose, i.e., repurposing the data. Researchers need not be aware that their reuse is a new purpose for this data. Regardless of the researcher's awareness of this, with a new purpose comes a different set of requirements and a different design for data organisation.

Sometimes a researcher's intended use of a data set and the purpose for which the data set was made align. In this case the scientist can use the data set for their purposes with minimal effort. More often, the intended use of the researcher and the original purpose of the data set do not align. Because of this preparing, curating and integrating data sources has become a primary task of e-scientists.

Repurposing of data allows the reuse of already existing data sets. This will allow the combination and analysis of this data in novel ways to answer questions that could not be answered before. Additionally, repurposing of data will also allow for faster and cheaper research, since already collected data can be reused to answer new questions.

Yet with all these data sets, and the prospect of answering new questions, e-scientists often struggle with these activities. In bioinformatics, it is believed that "fiddling with the data" may often consume more than half of the time of a Ph.D. project.[1]

[1] Personal communication with Prof.dr. A.H.C. van Kampen, head of the Bioinformatics Laboratory of the Academic Medical Center (AMC) of the University of Amsterdam.

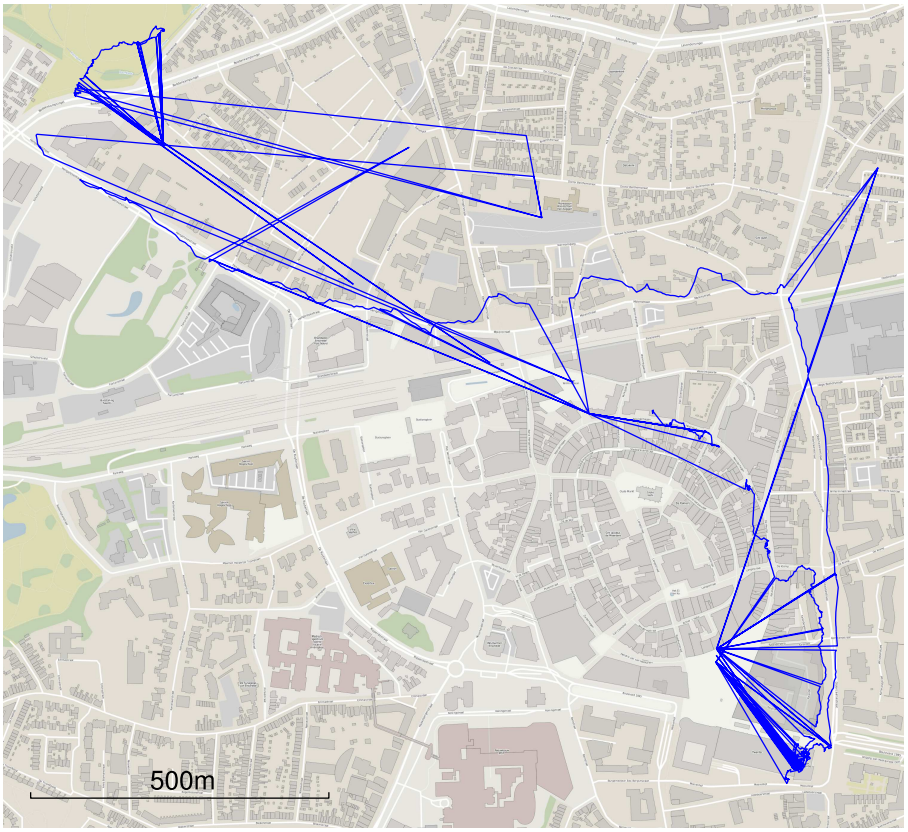


Figure 1.1: Position trace that can easily be mistaken for GPS trace from a mobile phone showing strange 'attractors', reprinted from [41].

5 Illustrating the struggle By reusing data for another purpose, one may encounter many unexpected, often subtle, problems with the data. See for example Figure 1.1 which depicts what, at first glance, seems to be a GPS trace from a mobile phone which appears to contain strange 'attractors', points where the position seems to bounce back-and-forth from. It may take some thinking and effort to find out that these are the locations of GSM cell towers: apparently when the GPS signal is lost, this phone's software reverts to the

nearest GSM cell tower position as a next-best position estimation.

Observe that the data presented in Figure 1.1 is not a GPS trace at all. It is a *position* trace. This semantical difference is the cause of the wrong assumption underlying the difficulty in discovering why these 'attractors' are present. The GPS trace semantic creates the expectation that without a GPS lock the position value would be missing, while being a *position* trace the value is determined by other means if no GPS data is available. This assumption leads to a data quality problem where the new purpose of the data requires non-GPS locations to be filtered out.

Furthermore, the danger is always present that 'nitty gritty' problems that are not discovered render results invalid. For example, [122] warns fellow bioinformaticians that analysing microarray data sets with Excel corrupts the data with automatic format conversions misinterpreting gene names for dates and Riken identifiers for floating point numbers. While the superficial problem might seem to be Excel's overzealous format conversions, the real underlying problem is the mistake in semantical interpretation and the lack of transparency about the interpretation and consequent (automatic) actions performed on the data.

6 The impact of data quality problems In enterprise information systems and business analytics, many reports can be found that highlight the importance of good data quality and how hard it is to obtain it. Dirty data costs US businesses billions of dollars annually and it is also estimated that data cleaning, a complex and labour-intensive process, accounts for 30% to 80% of the development time in a data warehouse project [11]. Key findings of a 2011 Gartner report [36] are:

(a) "Poor data quality is a primary reason for 40% of all business initiatives failing to achieve their targeted benefits,

(b) data quality affects overall labour productivity by as much as a 20%,

(c) as more business processes become automated, data quality becomes the rate limiting factor for overall process quality."

Although these numbers do not pertain to e-science, there is no reason to believe that they would be significantly more favourable.

Repurposing data concerns selecting data sources, extraction of data of interest from these sources, transformation to a target structure, cleaning data, coupling data from different sources that in some possibly novel way belong together, etc. It can be observed that e-scientists often struggle with these activities. Seligman et al. studied where time goes during data integration [99]. Although the study was broader than e-science, its conclusion most probably holds: not one of their seven categories of activities could be identified as the main culprit; they are all hard.

7 Cause of the struggle: quality and semantics The opening story of this chapter illustrates the context of the struggle to repurpose data. The scientist attempts to use a data source for a different purpose, and struggles to answer new questions with this data source without investing an enormous amount of effort.

It is our claim that the struggle to repurpose data is caused by problems with data quality, data source semantics and their interplay. A new purpose for data means different requirements on semantics and quality may be placed on the data.

8 The data quality struggle One often distinguishes many dimensions in data quality. We often speak of data quality along dimensions such as accuracy, consistency, completeness, currency, etc. [8]. Yet data quality is not evaluated in a vacuum. The dimensions of data quality are anchored and calibrated through the intended use of the data.

High quality data under one semantic may turn out to be usable as low quality data under other semantics, and vice versa. With the repurposing of data comes a recalibration of the data quality, which in turn might lead to the exploration of additional data sources to combine with the current ones to enhance the quality of the data. This, in turn, leads to repurposing these newly found data sources, and so on.

For example, think of data about melting points of materials published by multiple laboratories. Suppose that we want to combine and reuse this data not for the purpose of improving the melting points we know, but for the purpose of investigating the accuracy of measurements by each lab. These data sources feature some description of the measurement method, but the completeness and consistency of that data may be lacking. What was a group of high quality data on melting points is now of considerably lower quality: we need additional data about how and when these measurements were taken.

9 The semantics struggle To repurpose data and make it meet these new semantical requirements it is necessary to understand the current semantics. However, the published semantics of a data source, i.e., the semantics that are made public through documentation, differ from the actual semantics.

Published semantics associated with data sources often lag behind the actual developments, and thus the actual semantics. Even if the documentation is diligently kept up-to-date, the published semantics often lack the depth needed to fully grasp the meaning of the data.

The actual semantics of a data source are not something defined solely in a documented schema, but are defined by how fields and attributes are used by different persons. Unconsciously made assumptions by the creator of the data source create subtle differences between the published semantics as documented by the creator and the actual semantics as used by the creator.

Data sources created and curated by multiple authors have an additional layer of semantical complexity. Each author uses their own actual semantics. Even if the authors take care to use the same semantics, differences in interpretation of these semantics can lead to different actual semantics.

10 The interplay of quality and semantics When exploring the data source for repurposing the e-scientist seeks to uncover the actual semantics. The interpretation of the published semantics and the unconsciously made assumptions by the e-scientist play a role in how he conceptualises the actual semantics of the data source.

Peculiarities in semantics and quality are hard to discover and often found only by stumbling over them. The understood actual semantics lead to expectations about the data. Any violations of these expectations may uncover exceptional situations (semantics) or errors (data quality).

A misunderstanding of the actual semantics can lead to a harsh judgement of quality, even while the data source is of a high quality with respect to its intended purpose. The other way around, systemic error or a perceived pattern in the data can lead to a gross misunderstanding of the actual semantics. Both of these problems are compounded by data sources with multiple actual semantics.

11 Symptoms of the struggle Scientists are forced to manually 'massage' the data sets and make data integration decisions without the necessary information or insight. When they make mistakes, undoing unfortunate data integration decisions again takes time and manual effort. These inefficiencies prevent the scientist from getting results quickly, and reveal themselves through a number of symptoms:

- Wasted time through ambiguities, due to exceptions in the data, due to lacking or outdated documentation, and due to misunderstood and ambiguous actual semantics.
- Manual work on extraction, transformation and coupling because of a lack of tools for this job.
- Wasted time spent on redoing work because of errors in data use due to wrong assumptions about data quality.
- Redoing work due to the extraction of too much or too few data, and mistakes in data transformation and coupling conditions.
- Wasted time spent on backtracking from selected data sources due to new insights and discoveries.
- Data sources never seem to fit together, leading to spending a lot of time on aligning them.

An example of several of these symptoms can be found in [108], which investigates the quality of metabolic databases by reconstructing the well-known tricarboxylic acid (TCA) cycle from 10 human metabolic pathway databases. Consensus exists for only 3 of the numerous chemical reactions involved in the TCA cycle. While not reported, the work on this investigation and reconstruction of the TCA cycle from these 10 databases was largely done manually.

12 Lack of tool support E-scientists are forced to work in an inefficient way because of traditional assumptions about data integration and its goal. The traditional assumption of data integration is that the integration of data must be fully completed before data can be meaningfully used. This assumption forms the basis for many tools that support the Extract, Transform, Load (ETL) process of data warehousing.

E-science can be regarded as big data analytics for science. It differs from business analytics by among other things posing higher demands on quality of data and results. Science is about understanding and truth seeking, where rigour in method is needed to make sure that results really prove the claims. Furthermore, it is also different in that analytics for science is more explorative and unpredictable requiring a different way of working.

Because of the differences between business analytics and e-science, methods and technologies developed for business analytics do not fit well in the e-science workflow. At this moment, many of the symptoms of the struggle to reuse data are exacerbated by the lack of tool support for the way e-scientists work.

13 Aftermath of the struggle: no communication After a scientist has finally completed the task of integrating data from different sources they continue work towards their actual goal: getting research results. A side-effect of this research process is an increased understanding of the data sources, their original purpose, and the intricacies associated with reusing them. In other words, these discoveries are a valuable by-product of the process, which, however, are often not properly documented and shared.

Yet if this knowledge is not communicated to other researchers they will

have to undergo the same process of manually integrating the data source and building tools for doing it. It would be beneficial if any subsequent e-scientist working with the same sources would not have to go through the same painstaking process of data understanding. Moreover, documenting and publishing processing steps may better link a publication to its source data and will improve reproducibility [89].

1.2 Challenges

The concepts of quality and semantics are nebulous, and therefore difficult to formalise or make explicit:

- Quality is related to the original purpose of the data set. What is high quality for one purpose, can be low quality for another purpose. This form of quality is distinct from the quality of a data set's measured data: even if the data is measured diligently with the best equipment or the most principled collection methods it can have a low quality with respect to the new purpose.

- Semantics, in so far that they can be communicated, are equally difficult to make explicit. More effort has been put into doing so, and there are several frameworks for communicating semantics in a principled way, yet eventually they will still boil down to constructions based on natural language: semantics are determined by how the data is used.

Next to the twin challenges posed by the concepts of quality and semantics, the third challenge is the motivation behind current methods and tools. Methods and tools available for data combination aim towards 'traditional' data integration. They support difficult data integration scenario's but their aim is finding the single best integration, instead of assisting the data scientist in data understanding and combination with his established workflow.

1.3 Problem Statement

We pose the following problem statement:

> "How to support scientists in understanding data semantics and data quality to speed up data intensive research?"

In this thesis we focus on bioinformatics, a branch of biology research firmly engaged in e-science. The bioinformatics field is chosen because it is a good representation of a maturing data intensive field with non-trivial data. A lot of work in bioinformatics can be characterised as combining multiple data sources. An example would be the enhancement of measurements directly taken from the lab with additional annotations from algorithmic sources, or by combining new measurements with published data sets. For an excellent example of this see Section 1.7.3.

Based on the maturity and the presence of non-trivial data we assume that the approach taken in the bioinformatics field will generalise to other fields. As such, the same methods and tools can be applied to other fields with minimal adjustments. Other applications, such as business analytics, might require further adjustments depending on how these activities compare to the scientific workflow.

1.4 Direction and Research Questions

We propose to approach this problem in a two-pronged manner:

- The first part is a methodic direction to data understanding and repurposing. This includes the creation of methods and tools to improve the documentation of data understanding.

- The second part is a technical direction to the handling of data uncertainty. We propose methods and tools to for the integration of data through further automation, assuming the presence of messy data.

Both directions have the ultimate goal of providing the user with rapid feedback such that his possibly hidden assumptions can be challenged.

14 First direction: A method for data understanding and repurposing The traditional assumption about data integration is to first complete the integration and resolve all conflicts stemming from this integration. Science is about truth seeking and discovery, and with this comes a more erratic workflow when compared to business analytics. The lack of tool support exacerbates the struggle e-scientists experience when reusing data.

A valuable product of the process of data understanding and repurposing is an increased understanding of the data sources, their original purpose and the intricacies associated with reusing them. This knowledge is often not properly documented and shared, forcing other e-scientists to through the same hardships to reuse those sources.

A new method for data repurposing is needed that fits the e-scientist's workflow. Generic methods and tools can be developed that place the scientist in a central position, and adapt to the scientist's existing workflow. Furthermore, the e-scientist's workflow should be supported with regard to proper documentation of the data repurposing process.

Within this research direction we focus on the following research questions:

RQ1 "What is a good method for data understanding, data repurposing, and data analysis?"

RQ2 "What tool support is a natural improvement of the documentation activities in an e-scientist's existing workflow?"

15 Second direction: Technical handling of data quality Data understanding is a process, and ambiguous data will lead to initial assumptions being questioned. Because of this methods and tools that support data understanding should do so with an iterative approach. Initial assumptions can turn out to be false, requiring the user to go back and change the way they combine the data. Furthermore, data repurposing interacts with data quality:

the purpose with which the data was collected dictates, in part, the use of the data and the quality of the data.

We propose to model data quality problems and data combination choices as uncertainty [75, 60]. Using uncertainty in this way provides the option of leaving such data quality problems unresolved, while allowing meaningful use of the partially integrated data. Furthermore, the scientist is no longer forced to pick the single "best" integration option. He can express both option as being uncertain, postponing the actual choice while keeping the data usable.

Within this research direction we focus on the following research questions:

RQ3 "What is a generic foundation for uncertain data management that fits the method of **RQ1**?"

RQ4 "How well can the foundation from **RQ3** be applied to a bioinformatics use case using existing probabilistic data management technology?"

16 Engineering approach Secondary goals of this thesis are to place the tools generated for this approach in the open source domain, and to craft these tools at a minimum level of usability that renders them beyond mere throw-away proof of concept implementations. Tools that are generated and released should have appeal beyond supporting the experiments of this thesis.

17 Validation Validation of the work is done in two forms: a practical implementation of the theoretical framework, and a validation of the modelling of data quality problems as uncertainty on a real-world bioinformatics data set.

1.5 Contributions

The major contributions of this thesis are summed up by the following five items:

1. An iterative process for data understanding, data repurposing and data analysis (Chapter 2).

2. The design of a digital lab notebook using semi-structured data and supporting a quality guarding process (Chapter 3).

3. A 'data model'-agnostic framework for the definition of probabilistic databases (Chapter 4).

4. Validation of said framework on different data models (Chapter 5).

5. An application of the iterative process and the framework's principles on real-life data: grouping data (Chapter 6).

1.6 Related Work

As indicated, data preparation and integration may consume most of an e-scientist's time. There is a dire need for advancements in database technology to reduce this "data fiddling time" thereby rendering them much more productive. In this section, we will take a closer look at several areas of database technology and assess how well they support the e-scientist in his struggle with semantics, quality, and the e-science process and what advances are needed.

18 Data quality Data sources, or parts of data sources, of lesser quality may bring the overall quality of the integration results down [30].

Data quality measurement. There is some work on data quality measurement such as [26, 31, 82] measuring the trustworthiness of data sources, or [79] measuring the quality of rule based information extraction, but data quality measurement is largely an open problem.

Additionally, more research on data profiling is needed to allow for faster discovery of peculiarities, i.e., for a faster data understanding [1]. To enable true validation of such technologies measuring the overhead of data understanding for different degrees of repurposing is needed as well.

Semantic duplicates. A central data quality problem are semantic duplicates: two or more records that actually represent the same real-world entity. The goal of data integration is often to bring together data on the same real-world

entities from different sources. A straightforward approach may be thwarted by data sources copying from each other; automatic copy detection is needed [29].

There is much work on duplicate detection, also called record linkage, entity resolution and object identification [32]. But when confronted with real-world data, one quickly understands that more is needed. For example, granularity of entities may increase complexity: a large supplier is present both as firm "X" as well as "X Europe" and "X Asia". If updates in sources need to be incorporated frequently, an iterative approach to entity resolution is needed [117].

Information extraction from unstructured sources. Increasingly valuable data is embedded in unstructured sources. Therefore, the field of information extraction becomes ever more important. Whether harvesting data from web sites (e.g., [102]) or from social media messages (e.g., [45]), one thing is certain: natural language is inherently ambiguous [96], hence the extracted data is inherently noisy. Other types of more-or-less unstructured sources such as audio, video, or GPS traces may be even more noisy [15].

Data cleaning. Automatically repairing any problems in your data is of course an attractive prospect. For example, data imputation, filling in missing values with some kind of prediction, can — if done properly — improve analysis results in certain circumstances [104]. Nevertheless, data cleaning remains a hard problem both in terms on how to do it as well as on assessing what the consequences are for any subsequent analysis. Advances in data cleaning may, however, have significant impact as "analysts report spending upwards of 80% of their time on problems in data cleaning" [44].

Uncertainty in data integration. One important development with high potential for effectively handling data with problems is uncertain data. A good survey on uncertainty in data integration is [75]. In essence, the approach is to model all kinds of data quality problems as uncertainty in the data [60]. Uncertain data can be stored and managed in a probabilistic database [24, 56, 83].

Note that not only probabilistic databases can handle uncertainty in data. There are other models of representing uncertainty: the possibilistic or fuzzy set

model [121], and the Dempster-Shafer evidence model [100]. Furthermore, there are many different kinds of integration and data quality problems that deserve a probabilistic approach. For example, a semantic duplicate is almost never detected with absolute certainty unless both records are identical; a probabilistic database can simply directly store the indeterministic deduplication result [88].

19 Semantics Data understanding is primarily about uncovering the semantics of data in the data sources.

Data exploration. [23] describes the concept of *conditional functional dependencies*. The various kinds of functional dependencies specify constraints, or rather expectations, that are imposed on the data. Violations of these constraints may uncover exceptional situations (semantics) or errors (data quality). Functional dependencies can be mined from the data itself [1]. Such technology has much potential as it quickly gives both valuable insight into the semantics of data in a source as well as quality problems.

Other forms of data exploration are important for similar reasons. Techniques like exemplar queries [80] can be very useful for making a start with understanding a source: if you do not know much about the schema of a source, this technique can help you find data by giving an example of what one expects is in there, which when found gives clues as to how the example is represented in the source.

Another angle in uncovering the semantics of data, is to use the web to find (other) candidate terms for certain columns and tables in a source. The work of Google on Web Tables, where they harvest tabular data from websites including metadata, can perhaps be more widely used for this purpose [110]. Moreover, technology that exploits knowledge bases such as Yago [109], Wikidata [112], and DBPedia [7], for data understanding may be useful.

Answer explanation. The aforementioned techniques for data exploration are important, because the earlier one uncovers the true semantics of source data with all its peculiarities the better. Nevertheless as argued earlier, many peculiarities in semantics are found later in the process: one is confronted with strange (intermediate) results and asks the question “Why”. This is the field

of *answer explanation*: providing meaningful and useful reasons why certain answers are or are not in a query result [50]. One can also view this problem as attempting to find the *cause* of an answer being in the end result [78]. Answer explanation should be viewed broadly: also providing explanations for, for example, entity resolution decisions or other kinds of relationships between entities, is of great value for data understanding [33].

20 The e-science process The discoveries about the data embedded in the notes of an e-scientist are a valuable by-product of the process.

Data annotation and documentation. Documenting and publishing processing steps may better link a publication to its source data will improve reproducibility [89]. Since discoveries in data understanding are *about* data, effective means of referring to individual data items as well as specific subsets or slices of data, is needed. Although the fields of lineage and data provenance include data annotation techniques [18, 39, 12], to our knowledge such techniques have only sporadically been used to document discoveries made in data concerning data quality or semantics (e.g., [49]).

21 How this thesis contributes As argued here, many useful methods and techniques exist, but we have also given indications that in all areas there is a desire for more advances. The research directions of this thesis will contribute to several areas. The first research direction is aimed at improving the e-science process especially on the mentioned topic of documentation. The second research direction aims at improving probabilistic database technology which in turn allows important advances in almost all areas of data quality and semantics.

Furthermore, many techniques exist only in theory or as research prototypes. An e-scientist is only helped if the technology is at a sufficient Technology Readiness Level (TRL) to be used. Our engineering approach is directly aimed at addressing this issue by explicitly striving for tools of a maturity level higher than mere research prototypes.

1.7 Examples

Throughout this thesis we use a few examples to illustrate the concepts and perform experiments. The rest of this section elaborates on the examples of Named Entity Extraction and Disambiguation (Section 1.7.1), Maritime Evidence Combination (Section 1.7.2), and the Combination of Homology Databases (Section 1.7.3).

1.7.1 Named Entity Extraction and Disambiguation

We use natural language processing as a running example, the sub-task of Named Entity Extraction and Disambiguation (NEED) in particular. NEED attempts to detect named entities, i.e., phrases that refer to real-world objects.

22 Uncertainty through ambiguity Natural language is ambiguous, hence the NEED process is inherently uncertain. The example sentence of Figure 1.2 illustrates this: "Paris Hilton" may refer to a person (the American socialite, television personality, model, actress, and singer) or to a hotel in France. In the latter case, the sub-phrase "Paris" refers to the capital of France although there are many more places and other entities with the name "Paris", e.g., see Wikipedia [118] or a gazetteer like GeoNames [38].

23 Kinds of ambiguity A human immediately understands all this, but to a computer this is quite elusive. One typically distinguishes different kinds of ambiguity such as [69]:

(a) semantic ambiguity (to what class does an entity phrase belong, e.g., does "Paris" refer to a name or a location?),

(b) structural ambiguity (does a word belong to the entity or not, e.g., "Lake Garda" vs. "Garda"?), and

(c) reference ambiguity (to which real world entity does a phrase refer, e.g., does "Paris" refer to the capital of France or one of the other 158 Paris instances found in GeoNames?).

"Paris Hilton stayed in the Paris Hilton"

	phrase	pos	refers to
1	Paris Hilton	1,2	the person
2	Paris Hilton	1,2	the hotel
3	Paris	1	the capital of France
4	Paris	1	Paris, Ontario, Canada
5	Hilton	2	the hotel chain
6	Paris Hilton	6,7	the person
7	Paris Hilton	6,7	the hotel
8	Paris	6	the capital of France
9	Paris	6	Paris, Ontario, Canada
10	Hilton	7	the hotel chain
	⋮	⋮	⋮

Figure 1.2: Example natural language sentence with a few candidate annotations [61].

We represent detected entities and the uncertainty surrounding them as annotation candidates. Figure 1.2 contains a table with a few annotation candidates for the example sentence [61].

24 Dependencies between disambiguation candidates NEED typically is a multi-stage process where voluminous intermediary results need to be stored and manipulated. The dependencies between the candidates should be carefully maintained. For example, "Paris Hilton" can be a person or hotel, but not both, and "Paris" can only refer to a place if "Paris Hilton" is interpreted as hotel. We believe that a probabilistic database is well suited for such a task.

25 NEED is repurposing In effect using natural language processing to disambiguate and extract named entities is a form of reuse and repurposing. The data, i.e. sentences, is originally meant as a means of communication from one person to another, where both are presumed to have the same background knowledge and context. Reusing these sentences to extract relations between entities and to use those relations for analysis or understanding the sentence is a repurposing of these sentences for a new goal.

1.7.2 Maritime Evidence Combination

The second of our three running examples, the maritime evidence combination case is taken from real life. The maritime evidence combination case is published in [46].

Every day, a large number of vessels seek to enter the harbour of Rotterdam. One of the tasks of the coast guards is to ensure that vessels that attempt to smuggle goods into the harbour are stopped. Sending out patrol vessels to all incoming cargo vessels is infeasible due to time and cost constraints. Because of these constraints the coast guard must continuously make judgement calls on where to assign their resources to investigate those cargo vessels most likely to be smugglers.

26 Combining data sources To help the coast guards decision makers, it is required to integrate data coming from wide range of sources and reason over such diverse data. This work is done in the context of combining various data sources for integrated maritime services.

Data source are, for example, (i) Automatic Identification System (AIS), (ii) ship and voyage information, (iii) satellite/radar data, (iv) surveillance systems, and (v) coast guard reports.

Note that the reuse of data from many of these sources is a form of repurposing. Most of the data from these sources is not collected specifically to be used to investigate smuggling. Furthermore, many of the reports found in these data sources are in natural language, requiring natural language processing before they can be used automatically.

27 Uncertainty in knowledge Data in these sources may be incomplete and ambiguous. For example, according to VesselFinder [111] there are, at the time of writing, six vessels called “ZANDER”. For all but two of them, the International Maritime Organisation (IMO) number is missing. The IMO number is a unique reference for the ship. It should be manually entered at the time of installation of AIS on the vessel.

The IMO number might have been entered incorrectly [47], either by accident or with the intent to mislead. Alternatively, the knowledge-base can also be incomplete. As such, missing and imperfect information needs to be considered while evaluating any situation.

28 Uncertainty in observations If a coast guard reports a vessel called "ZANDER" by the coast, this does not precisely identify the ship. Since there are six vessels called "ZANDER", i.e., it is uncertain to which vessel the report belongs. Without any further information, the probability that the observed vessel is one of the six ships is $\frac{1}{6}$. However, this is a local view of the current situation.

When taking into account previously observed facts, we may derive a more accurate picture about the current situation. For example, if there exist prior report that a vessel called "ZANDER" sank, and another one was observed recently in some distant location, possibly with more identifying information such as an IMO-number, this evidence indirectly provides a more accurate picture on which ship "ZANDER" is observed by the coast guards.

The maritime evidence case as described in [46] has as ultimate goal the automatic determination of the chance that an observed vessel is engaged in smuggling based on a observations about these vessels.

29 Observational reports A large volume of intelligence reports, regardless of their origin, come in as text intended for human consumption. Natural language processing is used to extract facts and relations from the reports. As stated in Section 1.7.1, the named entity extraction and disambiguation stage of natural language processing handles voluminous intermediary results where the dependencies between candidates should be carefully maintained. A probabilistic approach to the handling of candidates allows facts and relations to be annotated with uncertainty.

Next to the uncertainty inherent in the natural language processing stage, there is the issue of trust: when receiving observational reports from data sources, how much weight should we give these reports? For example, if the

one of the data sources is known to automatically generate reports with older hardware, hardware that is known to produce false positives during periods of cold, the reports are still usable but the coast guards will trust the system less during winters.

In this thesis we focus on the representation of observational reports after the natural language phase, when the reports are represented in format intended for machine consumption state.

1.7.3 Combining Homology Databases

The final running example is the real-world bioinformatics case of combining homology databases containing groups of homologous proteins.

The main goal of homology is to conjecture the function of a gene or protein. Suppose we have identified a protein in disease-causing bacteria that, if silenced by a medicine, will kill the bacteria. A bioinformatician will want to make sure that the medicine will not have problematic side-effects in humans. A normal procedure is to try to find homologous proteins. If such proteins exist, they may also be targeted by the medicine, thus potentially causing side-effects.

30 The fictitious Paperbird taxa Orthology is one of the two homologous relations. We explain orthology, and orthologous groups, with an example featuring a fictitious paperbird taxa (see Figure 1.3). This fictitious taxa will be used throughout the thesis when referring to the homology case.

The evolution of the paperbird taxa started with the Ancient Paperbird, the extinct ancestor species of the paperbird genus. Through evolution the Ancient Paperbird species split into multiple species, the three prominent ones being the Long-beaked Paperbird, the Hopping Paperbird and the Running Paperbird. The Ancient Paperbird is conjectured to have genes $K\ L\ M$. After sequencing of their genetic code, it turns out that the Long-beaked Paperbird species has genes $A\ F$, the Hopping Paperbird species has genes $B\ D\ G$, and the Running Paperbird species has $C\ E\ H$.

For the sake of the example, the functions of the different genes are known

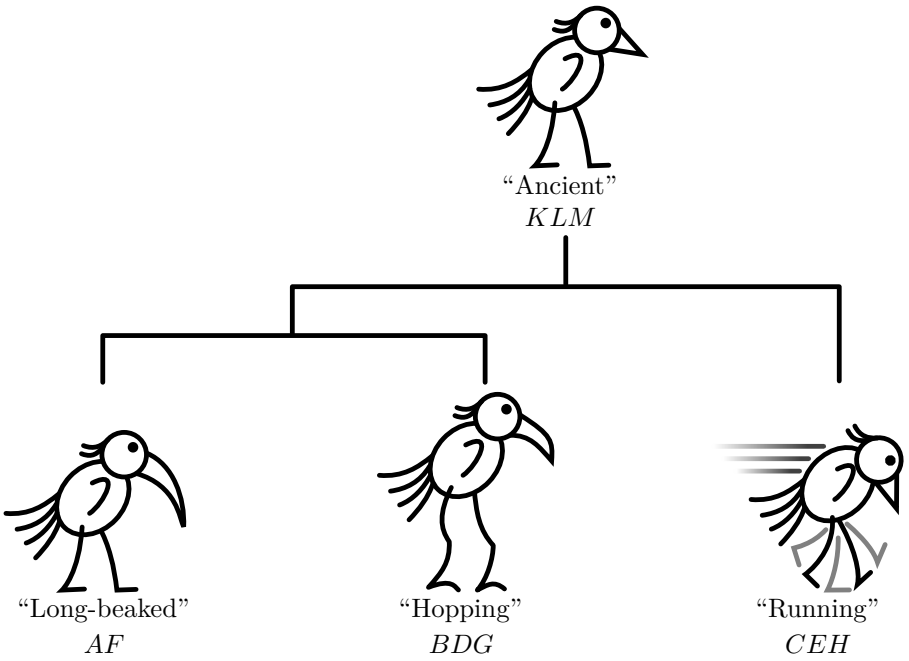


Figure 1.3: Paperbirds, hypothetical phylogenetic tree annotated with species names and genes.

to the reader. With real taxa, the functions of genes can be ambiguous. For the paperbird species, genes A, B and C are known to influence the beak's curvature. D and E influencing the beak's length. Finally, genes F, G and H are known to influence the flexibility of the legs. As can be deduced from Figure 1.3, these gene sequences are not complete. For example the Long-beaked Paperbird clearly has an elongated beak without having a gene to encode this quality.

31 Orthologs Genes D and E are known to govern the length of the beak. Based on this, on the similarity between the two sequences, and on the conjectured function of the beak curvature function ancestor gene L, we call D and E orthologous, with L as common ancestor.

Orthology relations are ternary relations between three genes: two genes in descendant species and the common ancestor gene from which they are evolved. The common ancestor is hypothetical. An orthologous group is defined as a group of genes with orthologous relations to every other member in the group. In this case, the group DE is an orthologous group.

How proteins are formed in an organism is largely dependent upon their genetic material. This leads to genes an proteins changing in similar ways during the evolution of a species. Therefore, proteins can by analogous arguments also be called orthologs. An extended review of orthology can be found in [67].

32 Paralogs A distinction commonly made is that between orthologous and paralogous proteins. Whereas an orthologous relation between proteins is established through speciation (the formation of a new species), paralogous relations are established through duplication. Looking back at the paperbird example, suppose that L is duplicated into L' and L'' in the Ancient Paperbird before it splits into two species. The Hopping Paperbird then features D' and D'', and the Running Paperbird features E' and E''. The relation between D' and E' is paralogous.

33 Creating homology databases There are various computational methods for determining orthology between genes from different species [72, 4]. These methods result in databases that contain groups of proteins or genes that are likely to be orthologous. Such databases are often made accessible to the scientific community. In our research, we aim to combine the insight into orthologous groupings contained in Homologene [84], PIRSF [120], and eggNOG [91].

Automated combination of these sources may provide a continuously evolving representation of the current combined scientific insight into orthologous groupings of higher quality than any single heuristic could provide for other bioinformaticians to utilise. This automatic combination is a clear example of data reuse and repurposing. By combining the insights from different computational methods bioinformaticians can answer questions that could not be

answered before.

One of the main problems is to distinguish between orthologs and paralogs. Computational methods are scrutinised for the way they make that distinction. Databases may disagree over which genes or proteins form an orthologous group, which are paralogs, and what the hypothesised common ancestor is. The distinction between orthologs and paralogs is beyond the scope of this thesis. What is important for our investigation of the homology use-case is the way proteins are grouped in the different data sources.

1.8 Thesis Overview

This thesis is conceptually divided into two parts. In the first part, Chapters 2 and 3, we focus on methods and tools to support the process of data understanding and repurposing and the documentation of insights gained during this process. In the second part, comprised of Chapters 4, 5 and 6, we focus on expressing uncertain data. In Chapter 7 we summarise our results and look toward the future.

34 An iterative method for repurposing In Chapter 2 we propose an iterative method for data repurposing based on the principles of pay-as-you-go, good-is-good-enough and keep-track-of-your-stuff. The method is characterised by quickly iterating through the steps of analysis, exploration and feedback.

In Chapter 3 we investigate the practice of note taking through the lens of a traditional research laboratory to highlight the opposing desires of the scientist and the institute. Based on this contrast we sketch our approach to automated support for documentation. We present the Strata system that implements the building blocks necessary for this support. We validate the abilities of the system have by prototyping a lab notebook system for the Prometheus laboratory of the University of Leuven.

35 A framework for creating uncertain databases In Chapter 4 we revisit the foundations of probabilistic databases and propose a formal framework

based on describing possible worlds. The proposed framework is independent from the underlying data model and separates meta data on uncertainty and attached probabilities from the actual data.

In Chapter 5 we validate the data model orthogonality of our proposed formal framework by applying it to Datalog, XPath and Relational Algebra, yielding robust and expressive probabilistic variants of these data models. Moreover, in Chapter 5 we illustrate how the formal framework creates two broad categories of optimisations.

In Chapter 6 we propose a generic technique for combining grouping data from multiple data sources, and validate this technique by applying it to the Homology use case described in Section 1.7.3. In applying our technique, we follow the iterative method outlined in Chapter 2.

CHAPTER 2

A method for repurposing

Before proposing a new method for repurposing, it is necessary to understand what the current method is. Recall the homology case presented in Section 1.7.3. There are large databases with homology information in them, each derived through different computational methods. Combining these sources could provide new insights.

Now, let's assume we want to investigate homologues for (sets of) proteins of specific species, but we do not want to limit ourselves to a single prediction method. For a more general purpose, assume we want to construct a data set that can answer questions of this sort.

This research project will require investigating each of the different data sources and repurposing them for our goals. Employing the currently practised method for integrating these sources works as follows.

36 A sketch of a case-driven approach We find a domain expert for the repurposing project, someone with knowledge about the field and — if at all possible — someone with experience with these specific databases. We let the domain expert select the appropriate data sources to use for answering our question. This domain expert will then go through the effort of reviewing homologous groups of the (sets of) target proteins.

Much of his effort will be spent analysing the current situation, and then combining, splitting or rejecting groups that are conflicting between the multiple data sources. He does most of this work based on his intuition about both the

subject matter, the semantics of the data sources, and the trustworthiness of the specific information of these proteins in these sources.

Note that individual pieces of information (records) in these sources are based on data from different research groups, from different experiments, done with different equipment in different labs, curated by different people, etc., hence the trustworthiness of each record in a source can be different.

Overall the whole integration project can take between several weeks to months. The duration is impacted by the amount of data that is available and the amount of understanding the domain expert needs to build up about the data sources.

During the project, the domain expert might make some personal notes about unexpected values and discovered semantics of a data source. He does so with the intent of referring to them later on, to make the work easier if he needs to review an earlier integration choice down the road. The notes also help him if he needs to answer a similar question again for a different (set of) proteins.

37 A sketch of a general approach For a more general approach, assume that we want to construct a data set that can answer *any* question about homologous groups of (sets of) proteins.

We start out roughly the same. We find a domain expert for the repurposing project, with the same qualifications as for the case-driven approach. We let the domain expert select the appropriate data sources that need to be integrated into a single new source. The domain expert will then go through the effort of understanding the intricacies of each data source, and deciding how to resolve integration conflicts where the sources disagree in some manner.

In most domains, some tool support is available for exploring the data. In the case of homologous groups, the domain expert can turn to ProGMAP [71]. The approach taken by ProGMAP is not to integrate the data sources directly, but to assist the domain expert by providing visualisations and showing information form different sources together. This approach highlights the differences between data sources such that the domain expert can more easily

do the integration himself.

The domain expert has to contend with the same issues as in the case-driven approach: he relies on his intuition on the subject matter, the semantics of the data sources and the trustworthiness of the information in these sources. If possible, the domain expert will turn to manually automating some of the work by writing integration scripts specific to the new purpose he wants to use the data sources for.

Yet most of the effort for the general purpose requires the same kind of work and exploration. The general purpose approach the process of integration is simply a much longer process. Overall the full integration of the selected data sources will take between months and years.

At the end of the project, the domain expert writes a document outlining the semantics of the different attributes and objects in the integrated data and, if time permits, a short tutorial on how to use it aimed at non-expert users.

38 Our proposed approach The current ad-hoc approach to data repurposing is based on manual effort by the domain expert guided by his intuition. His integration efforts are focussed on improving his understanding of the data sources and manually resolving conflicts. Tool support is provided by the querying abilities offered by the web interface of the data source, if any.

The current approach to a general data repurposing is based on the same manual effort by the domain expert. The domain expert's integration efforts are focussed on understanding the data sources enough to manually create integration automation for the specific new purpose. Some tool support is available for most domains, yet tools often focus on displaying information instead of actual integration.

Before we can automate parts of the data integration and repurposing process, the process itself must first be re-envisioned in a more principled manner. Basing our data integration and repurposing method on well-defined principles gives the method a more clearly defined process. Through this clearly defined process we can see what steps of the process can be fully or partially automated.

We propose a data repurposing and integration method based on the principles of 'pay as you go' and 'good is good enough'. In Section 2.1 we will discuss these principles and related concepts in detail. In Section 2.2 we present our method, followed by a discussion of the necessity of good documentation in Section 2.3.

2.1 Principles

The two principles of 'pay as you go' and 'good is good enough' are related. Here we outline their meaning, and how they can be applied to the problem of data repurposing. Further, we also present the idea that you need to 'keep track of your stuff', which is a necessity for collaboration and the sharing of data.

39 Pay as you go 'Pay as you go' means that you only put in effort at the moment you move forward. In an ideal pay as you go process, one only has to spend time and energy on improving the situation when it is clear what needs to be done to move forward. This effort is then directly applied to actually improving the situation, without having to put in work because of tangential concerns.

A perfect example of the 'pay as you go' principle in action is database cracking [57]: instead of creating an index beforehand, data is inserted into a table in an append-only style, which requires very little effort. Every query reorders the data in the table just a little bit or produces a little bit of indexing metadata just enough to answer the query, i.e., each query spends a little effort on the needed indexing. After many queries this sorts and indexes the whole table.

Being able to pay as you go requires that the work can be halted at any moment, while the progress so far persists and can be meaningfully used. One way to achieve this is to expend effort in small units by splitting up the necessary work in a sequence of, possibly repetitive, subtasks.

Persistence of progress means that the system should, after each small task, be stable and consistent. No unknown qualities should be introduced after any step.

When working towards the ideal situation, it is always possible to continue with a little more effort. The effort necessary to improve the situation becomes greater and greater, while the improvement becomes smaller and smaller. Knowing when to stop putting in effort is done by evaluating the situation through the 'good is good enough' concept.

40 Good is good enough When working towards a not necessarily perfect situation it is useful to know when to stop putting in effort. The idea of the 'good is good enough' concept is that you only put in the effort necessary to get to to a level that is (just) good enough.

The reasoning behind this principle is that any effort put in beyond getting to the good enough situation can also be used for other things. For example, think of the domain expert tasked with combining homology data sources. Let's say that he knows that the only questions that will be posed fit the "Do apes have an ortholog for the _____ protein in rats?" format. Given this pattern of questions, he knows that he is done when he has reviewed all homology groups that mention proteins from both rats and apes. He can stop and focus on another project until the moment someone tells him they are going to broaden the scope of their research.

As can be seen from the example, applying the idea of 'good is good enough' requires a definition of what is good enough. In the case of repurposing data, good enough is when the data can be used for the intended new purpose. So, to effectively apply the 'good is good enough' concept to one's work, one must have a clear idea of the new purpose and what good enough means for this purpose.

41 Keep track of your stuff Using the pay as you go scheme by taking small steps towards a situation that is good enough, we find that we frequently switch to other tasks that need doing. Every time we arrive at a situation that

is good enough, we start working on something else. And when we discover that our goals have shifted, as they are wont to do in both science and other endeavours, we come back to put in some more effort to move towards the new 'good enough'.

For an example, think back to the homology case described at the beginning of the chapter. Let us say that a new but similar question is posed to the domain expert, or that the initial answer (a set of homologous proteins) needs to be refined, e.g., the answer must be expanded with less reliable proteins, or restricted to only the most reliable ones, or additional information on the reliability of the obtained proteins must be added. In all these cases, the domain expert needs to retrace his steps, and having documented his work makes this much easier.

Coming back to something after a period of time requires reviewing our work. We look at the current situation and piece together how and why we are in this situation. By documenting our steps so far we can more quickly review the situation. We can look back at the record of choices that we have made and inspect our reasoning in the past. These notes are subjective, and based on our experiences with the data sources we are repurposing. Yet they contain valuable insights on the intricacies of the data sources, their semantics and the integration choices we have made so far.

If we keep track of all this information in an organised manner, we not only come back to the process more easily, we also unlock all this knowledge for others. As stated in Paragraph 13, effort in data understanding is wasted and repeated by others if not documented and shared properly. A well-organised ledger of notes, justifications for choices and insights is more readily sharable with others, leading to improved team work and collaboration.

42 Principles in Action: Ordering Food To illustrate the value of the above principles, we will show the difference between the traditional approach and our approach, based on principles through the analogy of ordering food.

We want to order food that is both tasty and cheap, which will be our definition of 'good enough'. Due to old-school advertising, we have access to a

big stack of price lists from several delivery places around town. A traditional approach would be to:

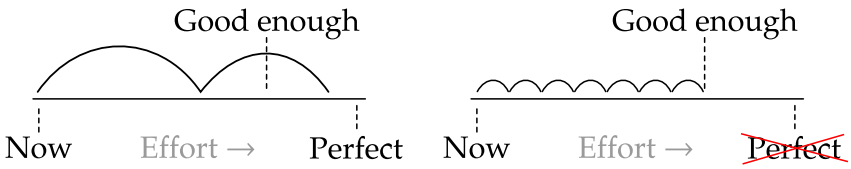
1. Gather all price lists,
2. merge them and make lists for rankings on both price and cuisine,
3. compare the prices and cuisines of all options,
4. place an order for the food that best fits our 'good enough' definition.

The traditional method makes sure that we place an order for the best possible food that meets our requirements. But we did so by spending a lot more effort than necessary to get to a food that is good enough — we had to work through all the price lists to find it. An approach based on the principles of 'pay as you go' and 'good is good enough' would be:

1. Look at the topmost price list,
2. is there a choice that fits our definition of 'good enough'?
   If so, skip to 4. If not, put it at the bottom, then continue on,
3. get another price list from the stack of advertisements, and go to 2,
4. order the food that fits our 'good enough' definition,
5. while waiting for the delivery, make a note of the found food. That way, next time you can immediately order it, and you can tell your friends about it.

Where the traditional approach has a large up front cost of merging all price listings, the pay as you go approach allows you to expend only as much effort as is necessary to find a food that fits the idea of 'good enough'.

43 Application of the principles Traditional data integration approaches feature a typical leapfrog behaviour as illustrated in Figure 2.1a. Once the work is started a significant amount of effort must be spend before arriving



(a) Traditional method. (b) Iterative method.

Figure 2.1: Diagrams of spent effort in traditional and iterative data integration methods.

at a situation where the data is usable again. Even if it is possible to do only part of the integration work because of the structure of the data, each 'jump' requires significant effort. This leads to wasted effort as the 'good enough' situation is passed by because all current integration conflicts must be resolved before the data can be used.

The problem of up front effort is illustrated clearly by the example of ordering food in Paragraph 42. Following the traditional approach as described requires spending a large amount of effort up front just ranking the items of all merged price lists on prices and cuisines. Only after this ranking is complete can the actual selection be made.

A traditional data integration approach is based on the evaluation of a 'good enough' metric over the whole situation. This is not necessarily an evaluation with full knowledge, but it is to the full extent of knowledge that is available. This typically means that all integration and cleaning is done before the data is used.

The effort needed for the pay as you go approach is illustrated in Figure 2.1b. Instead of big leapfrog jumps, this approach is characterised by many small steps forward, with the data being in a usable state after each small jump. This profile of spending effort in small steps makes it easier to hit the 'good enough' mark without overshooting. This is exemplified in Paragraph 42 with the 'pay as you go' approach to ordering food: each check of a price list is a single small step, and once a 'good enough' food has been found no further effort is needed.

The pay as you go approach requires evaluation of the 'good enough' metric based only on what has been seen so far and the estimation of what is still coming. In this way the 'good enough' metric takes on the characteristics of a heuristic.

With effort divided into many small jumps the 'pay as you go' approach meshes well with how scientists organise their work: they tend to make a decision based on incomplete information, and revise their opinions if new knowledge becomes available. While repurposing data, newly encountered data quality problems are analysed and if they do not impact the quality of the data for the purposes of the scientists, they should be postponed until they become an actual problem.

2.2 Process for Data Repurposing

Based on the principles discussed in the previous section we propose an iterative method for data repurposing. The design of the method has been informed by previous methods, our insights based on informal discussion with domain experts, and the global aim of this thesis to assist in the processes of data understanding and combination.

A comprehensive overview of previous data integration and data mining methods can be found in [76, 70]. Of these methods, our method has much in common with CRISP-DM [119, 101]. CRISP-DM is an iterative data mining process for use in industry with a focus on data modelling and deployment. In comparison, where CRISP-DM focuses on rigorous steps to produce a deployable model, our approach focuses on quick iterations and exploration.

Our method has a two-fold goal. Firstly, it is a method designed with scientific data repurposing in mind. The quick iterations allow the method to be applied even if the goal of the data integration radically changes, and the insights gained during the process are treated as a product. Secondly, it is also a method designed with tool support as an intrinsic part of the method in such a way as to make explicit where, and how, tools and technologies can assist in the process of data repurposing.

44 Requirements The principles we base our method on provide several clear requirements that the method, and the underlying process of data understanding and combination, must fulfil:

- To apply the principles of 'pay as you go' the process needs stability. This means that after each small step the data must be usable in a meaningful way.
- To apply the principle of 'good is good enough' we must be able to determine how well the combined data fits it's new purpose. A meaningful use of the combined data is evaluating the 'good enough' metric.
- To apply the principle of 'keep track of your stuff' throughout the process we must be build up documentation throughout the process. Additionally we have to be able to concisely refer to intermediate results and choices.

We address these broad requirements by expressing the results of the integration as uncertain data and by introducing the idea of a personal knowledge base.

45 Using uncertain data Expressing the results of an integration as uncertain data is an important direction with high potential for dealing with problems with data semantics and data quality [75]. It allows, for example, to postpone the resolution of such problems by modelling them as uncertainty in the data [60]. In essence, we gain the ability to use the intermediate results of the integration process in a meaningful way without being forced to resolve every issue in the integrated data.

Uncertain data can be stored and managed in a probabilistic database. In this way, an intermediate data integration result becomes a stable database in which certain data problems are properly represented and which can be readily queried and analysed.

As an illustration, an often occurring data quality problem are *semantic duplicates*, two or more distinct records in a database that actually refer to the same real-world entity. A semantic duplicate is almost never detected with absolute certainty unless both records are identical. Therefore, there is a grey

area of record pairs that may or may not be semantic duplicates. Instead of requiring a manual inspection and an absolute decision, a probabilistic database can simply directly store the indeterministic deduplication result [88]. Furthermore, the resulting data can be directly queried and analysed.

46 The personal knowledge base (PKB) The personal knowledge base is a repository of data integration knowledge. It contains rules for combining and integration, rules for what data we do not trust fully, discoveries that certain data is wrongfully produced as an intermediate result, discovered errors in the sources (and how to clean them), (wrong) assumptions on the semantics of certain attributes or subsets of data, regular transformations and data cleaning procedures, etc. The PKB is *personal* because the opinion on data quality and trust may differ from person to person.

In our method the integrated data is a derived product that can be automatically reproduced in full from the data sources and the contents of the PKB. Therefore, the knowledge in the PKB should be represented in an executable form. So, the actual product of data integration is the accumulated knowledge about the data sources, represented with integration decisions and rules in the personal knowledge base.

47 Meeting the requirements The stability requirement is met through application of uncertainty in data. Because one is not forced to resolve every issue in the integrated data, one can deal with them one-at-a-time. And by storing the integration result in a probabilistic database, the result of every small step is a database that can be used in a meaningful way. Together with the source data the PKB can be used to recreate a subjective personal view on the combined data.

The principle of good-is-good-enough requires an opinion on 'good enough'. This opinion comes from the domain expert who represents it in the PKB. One important meaningful use of an intermediate result is the ability to analyse the data to determine whether or not he current quality of the integrated data is good enough. Note that a good enough integration result may still contain

unresolved issues in the form of an acceptable amount of uncertainty in the data.

The intermediate integration result being meaningfully usable makes it more effective for discovering problems regarding data semantics and data quality. Such discoveries made by the domain expert are feedback which is represented in the PKB as choices made during the combination process, and rules that encapsulate discovered semantics for the data sources.

The knowledge base can be seen as a record of data integration choices that were made to arrive at the current results. As such, it also contains much information on the semantics and quality of the data sources. In its executable form, it contains concrete references to (sets of) data items in sources and intermediate results. Therefore, the PKB is a valuable source of documentation. The PKB can also be used as a basis for the reuse of effort through sharing and the creation of aggregate knowledge bases such as an 'institute knowledge base' to help new employees get started.

48 Steps of the process The method we propose consists of seven steps. An illustration of the method can be seen in Figure 2.2. The steps form two loops: the inner loop is the quick iteration loop (or the analyse-explore-feedback loop), and the outer loop captures changes in the definition of good enough. The steps of the method are as follows:

1. To start, the domain expert selects multiple data sources, each created with a specific purpose in mind. He makes a first initial and rough integration attempt by choosing rudimentary integration rules that will produce a partially integrated data set. This first integration attempt will have inconsistencies and semantic mismatches.

   The figure shows two data sources, one with three entities and data on three attributes for these entities, and the other with two entities and two attributes of which one attribute matches an attribute in the first source. The first attempt is based on a simple matching rule enabling the merger of data on entities that are semantically the same, i.e., refer

to the same real-world object: C and D in the figure.

2. The domain expert analyses the integration attempt to determine if the current level of integration is good enough for their purpose.

   The figure shows the domain expert, a bioinformatician abbreviated as BI, thinking through their usage of the data.

3. Typical usage of integrated data by the domain expert is exploring and querying to gain insight into the structure and properties of the data. The system should be able to explain the results of a query.

   In the figure, the user discovers that (a) although C and D refer to the same entity, the value of the shared attribute conflicts, producing an inconsistency (or rather uncertainty on the true value of the attribute), and (b) B should also match E but apparently the integration rules were not good enough yet to match and merge them.

4. Based on their analysis and exploration, the domain expert can issue feedback and improved integration rules to the system. These rules are then stored in the personal knowledge base.

   For example, he can decide to first provide the feedback that B should match E with which the system can refine the integration rules (possibly producing more matches than just B and E). Furthermore the choice of the domain expert is shown, with one option going back to step 3 and going through another iteration, and the other option moves on to step 6.

5. A new integration attempt is created in the next iteration, allowing the domain expert to continue refining their rules. This allows the domain expert to spend effort only when and where necessary.

6. When the integration level is to the domain expert's liking, i.e., when it matches his definition of 'good enough', he can stop iterating and use the integrated data.

In the figure, the final integration results are shown. The results are supported by the PKB, and the earlier feedback by the domain expert has been used to combine B and E.

7. If the definition 'good enough' changes, or if the inconsistencies cannot be solved good enough using the currently used set of sources, the domain expert can start the process from the top. He can select additional or different data sources, while keeping all the integration rules and choices in his personal knowledge base.

The personal knowledge base has a central role in the quick iteration loop where the analyse-explore-feedback cycle takes place. The mentioned 'system' shows where the domain expert benefits most from tools and technologies that assist him.

The domain expert spends most of his time iterating through the quick iteration. The rest of this section investigates the quick iteration loop, focussing on how the domain expert goes about his work, and how this work can be assisted.

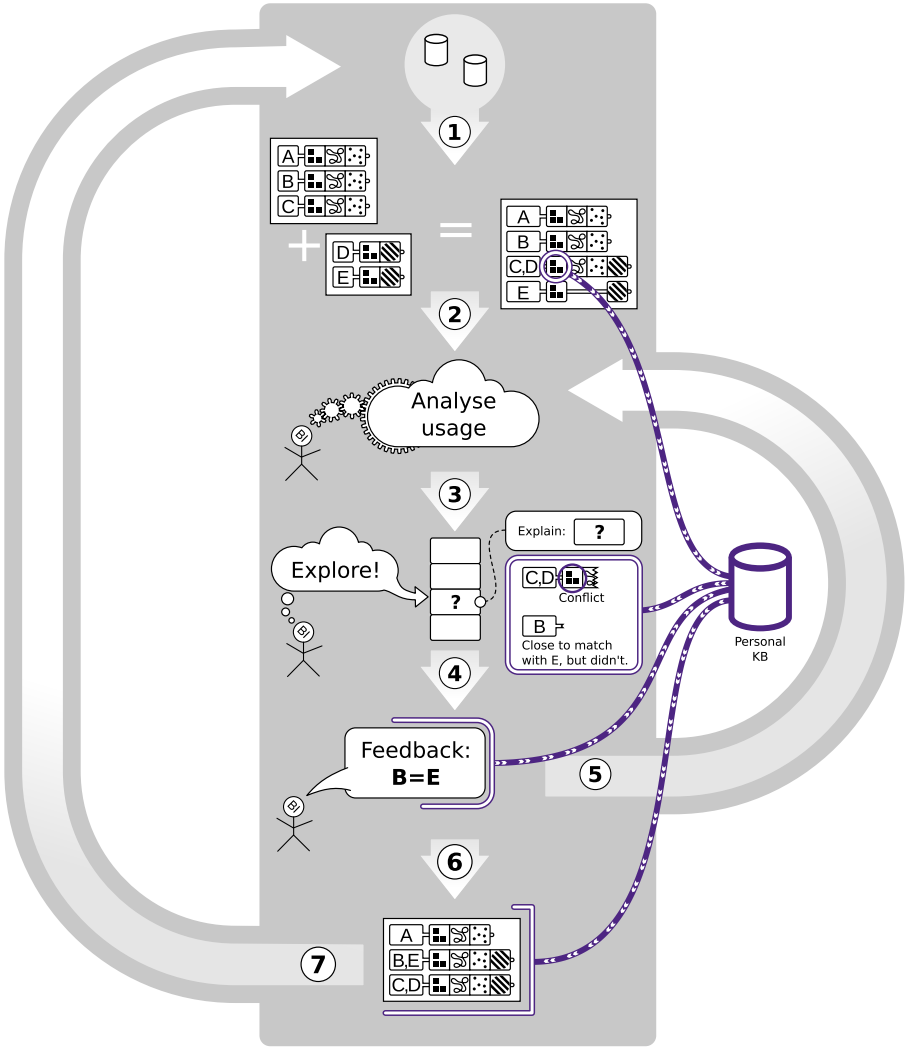


Figure 2.2: The steps in the iterative method for data repurposing.

49 Analysis The first steps in the quick iteration loop are the analysis and exploration of the integrated data. During the combination process both the input data and the combined data is analysed. Analysis of the input data increases the understanding of this data. This allows the user to refine his understanding of the semantics of the input data, possibly leading to an improved understanding of the semantics.

Analysis of the data may cast doubt on assumptions, or reveal that the scientist had hidden assumptions about the data. Exceptions or unexpected results can also indicate bad assumptions. An interesting example of such hidden assumptions comes from [105], the inspiration for the opening story (see Chapter 1) of this thesis. Imagine a research on pregnancy processes based on electronic patient dossiers (EPDs). Obviously one needs to extract those consult and treatment records that pertain to the pregnancies of the chosen population of women. It may easily happen that one only discovers many records not related to a pregnancy after obtaining the first analytical results. The assumption that all records of a pregnant woman during the pregnancy are related to the pregnancy is wrong: she may for example be treated for a condition she already had.

Analysis of the combined data helps determine if the 'good enough' quality is reached, and whether there are issues of trust or uncertainty that require feedback. With a sufficiently advanced system, this feedback comes in two kinds: voluntary and requested. Voluntary feedback is feedback given by the user without prompting by the system, for example as the result of discovering a pattern of data entry mistakes. Requested feedback is feedback that is given by the user after explicit prompting by the system.

50 Exploration to gain data understanding Improving the way the scientist gains an understanding of the data is a promising direction for the reduction of effort on data repurposing. Data understanding is a gradual process of exploration, revising one's idea of the semantics of a data source and running into the inevitable exceptions in the source's data. Unless the data sources are exceedingly simple, multiple iterations of refining one's understanding are

often necessary.

Exploration of the data source starts with getting one's bearings: querying some well-known elements, looking at a familiar aspect of the data source, or finding something of immediate interest to one's research. Based on this foothold, the scientist starts looking at how to use the data, what hurdles they need to overcome to use the data for their purposes. Again, this is done by inspecting the data through queries.

51 Exception finding An important aspect of data understanding is to discover the actual semantics, which may differ from the publicized semantics. The actual semantics of the data source are informed by both the commonly occurring patterns of data, and the exceptions on that pattern. Exception finding can be split into roughly two parts: discovering leads, and investigating those leads.

Once a lead is discovered, whether through automated means, by purposeful manual querying, or even by accident — the utterance "That's funny. . . " comes to mind — the lead is investigated. Or, alternatively, the investigation can be skipped by adding a distrust rule to the system telling it to distrust the lead, to distrust the data source it came from, or even to distrust the organisation that provided the data.

Through this investigation the scientist gains a deeper understanding of the data source. The lead can be an actual exception, but it might very well be an instance of a regularly occurring pattern of data that the scientist did not expect or that was not present in the published semantics. In both cases the scientist's assumptions about the data are challenged and reviewed.

The discovery of exceptions in the data source can be partially automated, especially for numerical data there are good methods [53]. However, finding exceptions in more complex patterns requires more complex methods to find them, such as automatic classifiers, or the scientist looking at the data himself.

52 Feedback and iteration By exploring the combined data the user discovers choices that need to be made to improve the quality of the data.

There are several kinds of choices, like simply picking the best option from a set of alternatives, declaring a certain combination of data invalid, adjusting the trust rating of a source or subset of data from a source, and even postponing the choice until later.

Once a choice has been made the choice will be recorded in the personal knowledge base, and the source data is recombined with the new choice taken into account. The domain expert now has the option to evaluate the choice and to determine if it helps in getting to the 'good enough' state, or if the choice should be rolled back.

2.3 Free and Structured Documentation

The process of data integration produces a large amount of actions and choices. Each conflict between two data sources, each ambiguity, every judgement of trust, all these require that a choice be made. Even if the domain expert opts to postpone the resolution of the problem he has made a choice.

Some of these choices are trivial. But over time, as the domain expert populates his personal knowledge base, the number of obvious choices decreases. Furthermore, obvious is a subjective measure: not all choices that the domain expert deems obvious, are obvious to those that want to use the integrated data. Having documentation about these choices goes beyond being helpful. With the growth of data driven research, having such documentation becomes a necessity.

To get the most out of the process of data understanding and repurposing, the whole gamut of actions and choices made by the domain expert needs to be represented in full in the documentation of his work. To do this, a blend of normal note taking and actionable structured data is needed. The domain expert needs to be able to refer to data in (intermediate) data sets, and to run queries that determine the data that his documentation refers to. Furthermore, the domain expert needs to be able to run queries that transform, integrate or clean a (subset of) data in a certain way while all of these actions are documented as they are taken.

In the rest of this section we first review typical note taking behaviour of scientists and follow up with describing functionality regarding actionable structured data. In Chapter 3 we elaborate on this vision of documented data understanding.

53 Note taking by scientists Even with the far-reaching digitisation of data and the near ubiquitous presence of tablets, a scientist's desk is often the home of multiple written notes on simple paper. Note taking is a time-honoured tradition of nearly all scientists. The recording of one's thoughts in quick and short manner helps to gather these thoughts at a later moment.

While some notes may only contain a quick scribble informing the scientist to look up a paper, mail to a colleague, or buy some milk on the way home, other notes contain valuable insights and pieces of information.

When collaborating with others, or when working on a larger project, the sharing and organisation of these valuable notes becomes a task in itself. Some notes are overall insights, others are very specific to a data source, or even to some part of a data source. Deeper research into the veracity of the data routinely involves reading related papers and looking at other sources of data. Documenting these investigations is a requirement for collaboration, and helps to communicate the quality of the work in publications.

To assist the scientist in data repurposing also entails assisting the scientist in taking notes about the data.

54 Ad-hoc queries and result annotation Both exploration of the data and exception finding require that the data is queried. These queries are often posed as so called 'one shot' queries. These queries are not meant for reuse and are only of interest at this moment. One shot queries have an ad-hoc nature and are written with the intent to answer a question currently on the scientist's mind.

The ad-hoc querying of the data, together with the ability to make notes directly with the query and it's results gives the scientist the freedom to investigate the data in their own way.

By keeping track of the changes to the query the scientist, or her collaborators can review the query and the notes at a later time. Having the query and the notes in the same system prevents them from deviating. This does not prevent the notes from becoming outdated, but there is a history showing the evolution of the query and the notes together.

55 Tracking actions and intermediate results Actions taken by the domain expert are an integral part of the repurposing process. The process of repurposing requires transformation actions, cleaning actions, doubting actions, filtering actions, etc. All of these actions generate new intermediate results that the domain expert investigates through ad-hoc queries.

By storing the actions of the domain expert as actionable structured data alongside the free form documentation we enable the reproduction of the whole process of integration and repurposing. The documentation in the PKB itself will be enough to automatically derive all intermediate results and the final integrated result directly from the documented actions and mentioned data sources.

2.4 Conclusions

We have proposed a method for data repurposing based on the principles of 'pay as you go', 'good is good enough' and 'keep track of your stuff'. The method is characterised by quickly iterating through the steps of analysis, exploration and feedback. After each iteration, the integrated data is in a usable state with unresolved integration issues being expressed as uncertainty *in* the data.

The proposed method highlights opportunities where the domain expert can be assisted through tools and technologies. Several of these opportunities present themselves through the introduction of a personal knowledge base that contains the rules and choices built up by the domain expert over the course of refining the integration.

The tools and technologies needed to fully implement the assistance as sketched in this chapter require further development on several fronts. In the

rest of this thesis we will focus on the support of documentation (Chapter 3), and expressing integration issues as uncertain data (Chapter 4, Chapter 5, and Chapter 6).

CHAPTER 3

Semi-freeform note taking

Parts of this chapter have been published as [113].

In the previous chapter we discussed the process of data understanding and repurposing, and the relevance of documentation in this process. In our vision the documentation of this process is actionable, i.e., the documentation can be used to reproduce the results of the process. To gain insight into the activity of documenting research we turn toward an old and established form of note taking.

56 Note taking in the laboratory In the quintessential laboratory, scientists perform experiments by manipulating physical materials and observing the effects. Based on these observations the scientists revise, reject, and formulate theories. While doing their research, the scientists working in these laboratories maintain a lab notebook which contains descriptions of experiments, results, and notes.

There is a long history of established practice with regard to lab notebooks in traditional laboratories. In this chapter we look at note taking through the lens of such a traditional laboratory to gain insight into the processes at play, and how these affect documentation activities.

An example of such a laboratory is the Prometheus group. Prometheus is the skeletal tissue engineering at the KU Leuven. The ultimate goal of this research group is to apply their developed tissue engineering methods in a

clinical trial setup. The use case of the Prometheus group is further described in Paragraph 73 and discussed in great detail in [25].

Note-taking in the form of the laboratory notebook holds a special place in laboratories like Prometheus. Since the ultimate goal is application of discovered methods in a clinical trial, it is important that all experiments are recorded in lab notebooks for future reference. Medical committees reviewing the request for a clinical trial review this data to assess the quality of the experiments, to determine the extent to which researchers followed established procedures, and to audit provenance information and chains of custody.

Though aware of the relevance of strict note taking, researchers are only human. Mistakes are made, and the system accounts for them. Furthermore, when performing experiments researchers would sometimes like to go off on a tangent, following an interesting measurement, or an observation that intrigues them. This leads to some tension between the scientist's desire to explore, and the institutes wishes to meticulously record every step of the experiment.

57 Outlook In Section 3.1 we shortly discuss how traditional lab notebooks relate to the e-scientist's work and data repurposing. In Section 3.2 we present, on a general level, the scientist's process and the stake of an institute and how these create a field of tension within the business process of the research group.

We propose a compromise to alleviate this tension in Section 3.3, and continue with a discussion how automation can support this compromise, and a short list of requirements for a support system. We follow up with Section 3.5, where we present our prototype system to support documentation activities, and summarise how this system can be used in the Prometheus laboratory.

Finally, we conclude the chapter in Section 3.6 by looking back at the insights provided by looking through the lens of the traditional laboratory.

3.1 Laboratory Notebooks

Laboratory notebooks are a good example of the kind of note taking scientists do. During an experiment the scientist takes notes of their observations and

decisions in the laboratory notebook. This record of the experiment serves several purposes:

- First and foremost, it is a record for the scientist himself. Having these notes allows him to review the events during the experiment. When an unexpected result comes up, the notes can give clues as to what could be the cause of this peculiarity.
- The notebook also serves as a primary source of information for publishing results. The notes form the basis for a rational reconstruction of events and preliminaries that form a large part of the research.
- Lab notebooks are a valuable source of information for in-house collaboration between colleagues in the same institute. Scientists can use lab notebooks of their peers to find results of similar experiments, possibly to compare outcomes, or to discover more about an unexpected result. Past lab notebooks are also a good source of previous work: while most publications include notes on previous work, the lab notebooks typically contain much more elaborate descriptions and annotations.
- Finally, nearly all institutes enact data integrity policies or approved laboratory procedures. The principal role the laboratory notebook plays as a near-chronological record of the actual research make it a good candidate for forming the basis for these policies and procedures.

58 The lab notebook in e-science Analogous to the uses of a laboratory notebook of a scientist working on experiments in a lab, is the notebook of a scientist working on integrating data to repurpose it for new research: the notes are primarily a record for the scientist, while also serving as a way to communicate insights about the data.

In typical data intensive research, data from several sources is combined before analysis. Looking at data intensive research from a traditional point of view, one can see the core "experiment" being the analysis of the combined data. The data preparations, such as repurposing and integration, then are a natural

part of the experimental setup, and the notebook contains the scientist's notes on the quality and semantics of the used data sources.

Publications grounded in the fourth paradigm (see Section 1.1) will also need to justify their reuse of the data. In current publications this is done by providing an "experimental setup" section preceding the presentation and discussion of the experimental results. The experimental setup section contains a rough sketch of what sources were used, and how they were combined.

59 'Note taking' becomes documenting Lab notebooks are a formalised form of note taking as mentioned in Section 2.3. Most traditional institutes have guidelines and regulations on the handling of lab notebooks, and they are an integral part of the laboratory's business process. With the number of scientific disciplines that rely on large data sets growing we can expect institutes to start enacting new policies to ensure the scientific rigour and integrity of data use and reuse.

Give the success of the lab notebook in improving scientific rigour and accountability, the note taking done during data intensive research are poised to become one of the foundations of these new policies.

Moreover, the e-science notebook can fulfil a prominent role not only as a collection of notes, but as a source of documentation. Not only does the ideal e-science notebook contain the valuable research results, it also contains documentation on the semantics and peculiarities of the used data sources. The e-science notebook also documents the steps used to integrate these data sources, and can therefore be used in the verification of the enacted policies to uphold the integrity of data use and reuse.

3.2 Tension between Workflows

The business process of a research institute is formed by the interplay of the researcher's interests, and the interests of the institute. This section consists of two parallel segments. In both segments we highlight the relevance of note taking and documentation:

- In the first segment we investigate the creative process of the scientist, and the ideal research workflow that follows from this if the scientist is given the freedom to explore.

- In the second segment we follow up with an investigation of the stakes a research institute has in research, and the ideal workflow that follows from these stakes if the institute could fully prescribe a workflow.

In the closing of the section we juxtapose the two ideal workflows. This serves to highlight the field of tension between the workflows, and to identify the options for a compromise that satisfies both parties. The compromise itself is presented in Section 3.3.

3.2.1 Scientist

The work of a scientist is an inherently creative effort. While most scientists will agree that there is a lot of rote effort that goes into their work, the core work of research and science is unpredictable. A scientist does not know what they will discover upon starting their work.

Science is a creative effort, driven by ideas and inspiration. Scientists are the ultimate specialists — we gain the most out of their creativity by letting them guide their own work, follow their own workflow without constraint.

In this way, they resemble artists. Not only are practitioners of both disciplines well aware of the fact that mistakes can and will happen, but a single observation or intermediate result can lead to a fundamentally altered approach. An exploring scientist only looks one or two steps ahead, with only a few points on the horizon as guide.

60 The scientist's ideal workflow The workflow of an exploring scientist is one of quick cycles, jumps and backtracking. The scientist formulates a hypothesis (not necessarily in a formal manner), experiments, revises, and experiments again. Based on their observations and inspirations, they easily jump to new approaches and investigate results. After such a tangent, many

scientists backtrack to some previous idea to try variations, or to revise them based on a later discovery.

In practice, the scientist knows that standardised workflows are a useful tool. Scientists in some fields, such as the biomedical field, incorporate automated workflows in their own discovery. In other fields, such workflows are used to allow delegation of surveys and polls. But in all cases there are many reasons to deviate from this workflow: mistakes happen and need to be recovered from, or curiosity leads the scientist to investigate on a tangent.

When looking at note taking and documentation, each scientist has a system that matches their natural workflow of exploration and discovery. Every scientist has their own method of taking notes. Some come back to them later to add further notes, or to annotate the data based on new insights. They want to be able to make corrections in the future.

The scientist desires a note taking system that supports their own workflow, so it does not get in the way of their actual work. Their notes are organised in a way that matches their workflow, so they can quickly find the notes they need right now. The notes themselves are very efficient. They incorporate exactly enough information for the scientist to review them later, without putting in effort that could be used exploring.

3.2.2 Institute

The institute's stakes in the research performed under its auspices are shaped in part by the different ethics boards that set out guidelines for conduct. Ethics boards are responsible for determining guidelines for ethical conduct with regards to the handling of dangerous materials, biological agents and machines, personal or sensitive information, as well as judging the risks and susceptibility of dangerous applications of the research (such as weaponisation).

The research institute has several reasons to adhere to external guidelines encompassing scientific rigour, integrity and behaviour. Among these there are two major reasons for institutes to adopt guidelines:

- First, the ubiquitous need to secure additional funding for new research

projects drives an institute to improve research quality, and to adopt guidelines that align with the perception of scientific rigour and integrity. Attracting funding and performing rigorous research enhances the institute's reputation.

- Second, when studying sensitive data, such as medical data from hospitals, or personal digital footprints from online sources, it is necessary to obtain permission from ethics boards before a study is conducted. The research institute can adopt guidelines to guarantee privacy and anonymisation of data.

Given that there is competition between institutes for the limited financial resources, reducing start-up time for new research is crucial. Therefore, it is in the interest of the research institute to regard the notes generated during data preparation as a valuable resource to be used in future research.

61 The institute's ideal workflow Even though the institute also wants research to run smoothly, the institute desires a system that works in a fully structured way to enable external audits and quality assurances to be made without effort.

Standardized workflows are needed before ethics and medical boards sign off on proposed studies. These workflows include the provenance of research results, annotations and a scientist's notes in the process. These workflows need to produce clear audit trails to justify how and why they were changed. Further assurances on data quality and integrity are necessary before approval of medical trails.

The execution of standardized workflows needs to be logged in the exact same way every time, so as to enable reviews of the scientific quality. Furthermore, structured notes and annotations by scientists allow the creation of aggregate statistics necessary to determine whether to invest in new equipment or personnel.

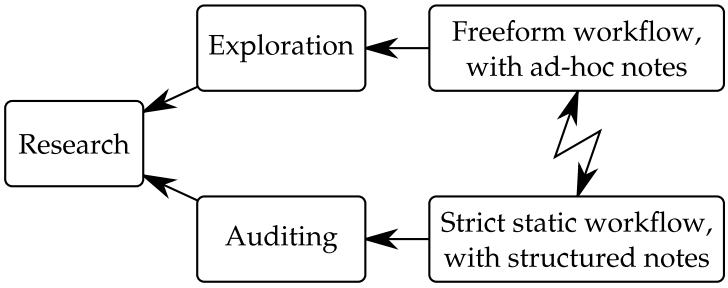


Figure 3.1: Tension between the scientist's and institute's ideal workflows.

3.2.3 Tension

Both the scientist and the institute have the objective of doing research. Looking at the two ideal workflows we known that: the scientist needs exploration to be able to to do his job, the institute needs audits to assure quality and integrity of the research.

Where the scientist wants a freeform workflow so he can do the work on his own terms, the institute desires a rigid workflow that facilitates audits of the work. Both parties want to perform high quality research, but the manner in which this outcome is to be reached differs for each. Both parties are aware of the relevance of the other's needs, but there is no obvious satisfactory solution. This situation is illustrated in Figure 3.1.

With regard to note taking, the situation is analogous. The institute wants to bind the notes to the workflow, keeping them unchanged from the moment they are recorded. In contrast, the scientists wants to write some notes down now, and come back to them in the future to revise or update them.

3.3 Compromise

In laboratories all over the world, a compromise has been established with regard to the ideal workflows of scientist and institute. This compromise is based on the inclusion of both experimental data and organisational data in

the physical lab notebook.

The experimental data consists of all the information associated with a single experiment. Such data includes tables, figures, recorded values, and annotations by the scientist. Raw data is typically analysed through statistical techniques, but the derived values are normally still considered experimental data.

The experimental data is complemented with organisational data. Organisational data describes the bigger picture. It describes why the experimental data was gathered, and in this way provides context for the experimental data.

The crux of the compromise is that scientists record the experimental data and organisational data in a *structured* manner, while recording their annotations and observations in an *unstructured* manner. At the moment, the balance of the compromise leans in favour of the institute as everyone involved understands the relevance of the audits and the necessity of review by ethics and medical boards.

62 A digital laboratory notebook In traditional research laboratories, such as the laboratories at Prometheus, scientists record their structured and unstructured notes in paper notebooks. This has significant shortcomings with regard to retrieving past experimental results from archives.

A single researcher working solo on a project will most likely find his recent results again. Over time, he accrues more and more paper notebooks, many of which end up in archives. If multiple researchers collaborate, they fill up notebooks even faster, leading to a faster growing archive. Investigating similar results is an exercise in browsing the archives by hand, and copying results for later comparison. Posing any kind of complex question involving data from multiple experiments is time-consuming and difficult.

Switching to a digital lab notebook resolves many of the current shortcomings of the paper lab notebook, and offers a way to improve on the current compromise for both the scientist and the institute.

63 Dichotomy of behavioural guideline automation The compromise described above is in effect a behavioural guideline. Scientists follow a certain set of rules and regulations to allow the necessary audits that the institute wants. From the first two chapters of [74] we learn that:

- Behavioural guidelines codified as rules and regulations in the physical world start from a basis of permissiveness, and work by constraining the actions of those following them. As such these systems, even though not always experienced as such by those who work within their constraints, allow individuals some leeway in how to fit them into their individual workflows. Furthermore, exceptions can be made on an almost regular basis. Handling of such exceptions is up to the governing body that set up the guidelines.

- Behavioural guidelines codified as instructions for a computer system traditionally start from a basis of constraint, only allowing those actions and processes that are actively programmed into it. In stark contrast with codified behavioural guidelines, computer systems are bound to only allow exactly those situations they were programmed for. Unless the programmers of the system took great care, a computer system will enforce the behavioural guidelines it was built for as if they were laws of nature, and violation of its constraints will simply be impossible.

Based on this insight we conjecture that this contrast of the permissive versus the constrained has created an atmosphere of annoyance and aversion when it comes to the automation of behavioural guidelines regardless of the organisation that wants to do so. Given the public history of automation of guidelines and regulations, the end user, that is, the person that is eventually going to work with the automated system, expects that using the system will force them to adopt a workflow that matches the system.

We conclude that a system that supports the workflow of the scientists will therefore gain greater adoption the more it matches both the scientist's established and the ideal workflow.

64 Sketch of an automated notebook system As discussed, a system that fulfils the desires of both the institute and the scientist requires compromise. Given the variety and individuality of the scientists, and the institute's ultimate desire of having productive scientists with high morale, we want to put the scientist's workflow in a central position.

Our approach to create an automated notebook system is based on the following three key features:

- Freeform note taking: the system supports a scientist's personal style of note taking and documentation by offering freeform note taking, the linking of notes and the option to add annotations on any topic.
- Eventual observation of policy: the system does not feature hard constraints that prevent the scientists from continuing their work, but instead presents policy information and warnings that apply to the context the scientist is currently working in.
- Continuous policy evaluation: taking full advantage of the fact that we can offload the effort of rote bookkeeping to a computer, the system should provide the scientist and the institute with immediate and automatic evaluations of adherence to guidelines and policies.

We can obtain a system with these properties in the same way that a compromise has been reached in the traditional paper lab notebooks: by mixing structured and unstructured information.

65 Presenting and evaluating policy By making the guidelines and organisational constraints known beforehand, in the same system scientists are using for their notes and documentation, the scientists are informed about the guidelines.

Instead of introducing hard constraints that prevent the scientist from entering their data and notes in the system, the system can continuously assess the scientist's compliance with the institute's guidelines and notify the scientist of found violations with reference to the guideline itself. From an organisational

point of view this still allows the monitoring of violations, while at the same time allowing scientists to enter data partially, acknowledge the violation, and come back later to continue the work and move back to a compliant situation.

Allowing the temporary violation of the guidelines, with the understanding that it can be resolved at a later point provides the scientist with breathing space. The freedom to deviate from the guidelines creates the opportunity to fit the organisational requirements and constraints into their own workflow.

66 Mixing structured and unstructured information Automatically assessing violations of the guidelines requires structured data that can be understood by the computer.

From the point of view of the institute the guidelines can be expressed in the system such that they are automatically assessed, giving the institute an up-to-date overview of compliance with the adopted procedures.

From the point of view of the scientist, the benefits of having structured data are much easier reuse of data from earlier notes, support from the system itself in interpreting and visualising experimental results, and the ability to pose complex questions over multiple experiments.

67 Team work, sharing and collaboration An additional benefit to both the scientists and the institute is the ability of scientists to make notes with the guidelines and policies, and immediately sharing these with their colleagues. In doing so, the guidelines and policies become a bigger part of the scientist's workflow, instead of being imposed from outside. Further, documenting practical matters related to policy decreases the time a new colleague needs to spend on learning how to deal with the regulations.

Another significant benefit of adopting a digital lab notebook is the ability to more easily share annotations, experimental data and organisational data. This benefit seems marginal at best to readers that are used to handling digital information. Yet this is a marked improvement over copying the physical lab notebook pages either by hand or with a photocopier.

Easy sharing within the same institute eases reviews and collaboration.

Teams can more easily work together on the same project without having to cross-reference pages from several physical notebooks, and it becomes easier to take over for a colleague when they are unavailable to perform a time-sensitive experiment.

The same qualities that improve collaboration within the institute also improve collaboration with others outside the institute. Experimental measurements and derived results can be shared through the digital system, and collaborators can view them remotely and annotate results. Discussion about experiments and organisational details becomes much easier due to the ability to directly refer to certain results: a simple link in an email suffices to make it perfectly clear as to what measurement one is referring.

3.4 Proof of concept: Strata

In this section we present the Strata system. Within Strata it is possible to give a structured description of organisational constraints allowing automated assessment, to have multiple users collaborate on documenting their work, and to mix structured data and unstructured data. Strata is based on the concept of a semantic wiki and extends from there with a novel type system.

The design of Strata was guided by the features presented in Paragraph 64. To show how Strata supports the creation of an automated lab notebook prototype, we align the key features for the automated notebook with Strata's abilities:

- Freeform note taking: the Strata system is based on the concept of a semantic wiki, and builds on top of the existing wiki platform DokuWiki [28]. A wiki is a web application that allows people to collaboratively add, modify, or delete content. Wikis have little implicit structure, allowing structure to emerge according to the needs of users. In general, wikis are centred around pages: typically content about a single topic resides on a specific page. See Paragraph 68 for further details.

- Eventual observation of policy: the Strata system's flexible data entry

and type hinting system allow the scientist to temporarily circumvent the constraints on regulated data entry, while being notified of the fact that the entered information is not yet correct. See Paragraph 69 for further details on data entry and querying, Paragraph 70 details how type hinting works, and Paragraph 71 discusses Strata's approach to notifications.

- Continuous policy evaluation: the consistency checks and data formats required by policy can be expressed in Strata in the same way other organisational data is handled. By describing the data formats in Strata, the query system can be used to describe consistency checks and create reports of current violations. See Paragraph 72 for further details on consistency checks.

Strata's underlying data model is the well-known Resource Description Framework (RDF), stored in a Relational Database Management System (RDMBS). All triples in the relational database management system are derived from the structured data on the actual wiki pages. In effect, the RDBMS' function is to serve as an index to speed up query answering.

68 A semantic wiki as base Because of the fragmented nature of the information on a wiki, it is hard to share content across pages. When multiple views on the same content are desired this content is usually added multiple times to the wiki. Repeated content leads to duplication of work and easily introduces inconsistencies.

To tackle this problem, a semantic wiki allows the entry of structured data, allowing (untyped) data to be used across pages. Whereas this avoids most inconsistencies, it still remains challenging to enter data consistently. When data entry is split across multiple pages, not all values might be entered in the same way. Therefore, it is beneficial to use data types, in order to see that both values represent the same data.

Several implementations of semantic wikis exist: Semantic MediaWiki [68], IkeWiki [98], SweetWiki [14] among numerous others. Of these, only Semantic MediaWiki sees significant use on publicly accessible sites. Most of these

```
<data person>
Full Name: John Doe
Birthday [date]: 1984-03-02
</data>
```

john_doe *(person)*	
Full Name	John Doe
Birthday	1984-03-02

Figure 3.2: Data entry with ‘date’ type hint.

implementations are research prototypes that implement their own wiki engine. This approach works well for the purpose of researching new methods and facilities. It works less well for the adoption of semantic wikis by the larger user base. By building on top of the well-known DokuWiki [28], we hope to increase the public’s familiarity with semantic wikis and promote adoption of semantic technologies.

The Structured Data plugin [40] is an effort that builds on top of DokuWiki, but it provides a much simpler data model, which lacks extensive query capabilities.

69 Structured data entry and querying Strata features a query language derived from simplified SPARQL and a custom data entry language designed for users from a wide array of technical and non-technical backgrounds. Design trade-offs of complexity versus syntax simplicity have been decided in favour of syntax simplicity, leading to a simpler language that omits several of the advanced uses of SPARQL.

Data is entered with a dedicated wiki syntax as a series of key-value pairs that is associated with a page or fragment. A key may have zero or more values. Type hints can be given to determine normalisation and display format of entered data, type hinting is detailed in Paragraph 70. An example of a simple data entry for a person can be seen in Figure 3.2.

Entered data will be coupled to the page it is entered on. As is common practice in URLs fragment identifiers, i.e., the part after the hash symbol in `page#fragment`, can be used to subdivide the data entered on a single page into smaller parts relating to different, typically related, subjects.

```
<list ?name ?birthday>
?p is a: person
?p Full Name: ?name
?p Birthday [date]: ?birthday
?birthday <= 1990-01-01
</list>
```

- Edsger Dijkstra (1930-05-11)
- Gerrit Blaauw (1924-07-17)
- Guido van Rossum (1956-01-31)
- John Doe (1984-03-02)
- Piet Beertema (1943-10-22)

Figure 3.3: Query to show a list of people born before the 1st of January 1990.

Strata does not enforce a specific structure for entered data, and allows the user freedom to choose what keys to include or leave out. Further, the Strata system makes no assumptions about the relation between fragments. Based on our observations most users interpret entered data as data about the page's subject, and thus expect fragments to be related to the page's subject as well.

Data querying can be used to create indices, overviews or summaries for human consumption. The user can query the data by describing the pattern of data that interests him, an example of which can be seen in Figure 3.3.

The query language is designed to match the syntax for data entry and to allow expressing simple queries in an intuitive way. Type hints can be used in the query language to associate types with variables and literals. Literals within the query, such as a date, are normalised according to their associated types. Types are propagated through the variables. In the example in Figure 3.3 the literal in the comparison is automatically normalised as a 'date' due to the hinted type associated with '?birthday'.

It is our experience that most users formulate their queries in an 'is this true?' fashion. Query answering matches this expectation by collapsing multiple identical answers into a single answer. In effect, query answering is done using set semantics instead of SPARQL's bag semantics.

The results of a query are displayed to the user in table, list or custom format. Both table and list formats support client-side filtering and sorting. The custom format is used to display the results as a sequence of template instances (see Paragraph 71).

Type hints in the query are used to determine appropriate display forms

for the resulting values in all three of the display formats. Simple aggregates can be used to display a count or sum of values.

70 Type hinting Instead of a full typing system, Strata employs what we call 'type hinting'. Type hints given by the user of the system are used in two ways:

- To normalise data during data entry, before the data is stored in the index in the form of RDF triples. If the entered data can not be normalised by the given type hint, the value is stored verbatim.
- To influence how data is displayed to the user, such as when the entered value is displayed on the page where the data was entered, or in listings and tables. If the type hinted at does not understand the normalised data, it shows the value in the index verbatim.

The type information is not stored with the data itself, but is handled purely as meta data at the interface of the system. For example, if some data is entered in the system with the type 'date', it is normalised to a timestamp before being stored. If it is later presented as the result to a query, but without any type hint, the normalised timestamp will be shown. For example, if one event is listed on *27-08-2014* and another one on *27 August 2014*, both events have a different date when the values are compared literally. The 'date' type normalises date values to timestamps that allow comparison, and displays them in the unambiguous year-month-day format [106].

By not forcing a strict typing system, the users of the system are free to deviate from the hinted type if this suits them better. This allows easier migration form one form of presentation to another, at the expense of not carrying typing information to other locations where the data might be used.

71 Templates and notifications Instead of showing certain notifications if some condition is matched, Strata provides the building blocks to allow the designer of the automated lab notebook to create their own notifications in a way that fits their policy.

Strata supports the definition of custom templates that describe how to display a set of values. An example of this would be a custom display for persons such as the John Doe entry in Figure 3.2. The user can use the already familiar DokuWiki syntax in the template and adds placeholders for the values.

The second use of templates is for the custom display of query results. As described in Paragraph 69, query results can be displayed using a custom view. Templates used to display data entries and query results do not differ from each other, allowing the reuse of templates to provide a coherent visual display for data of a single type, e.g., all person information can look the same everywhere in the system.

Notifications can be created through nesting of templates. For example, let's assume that the `Birthday` field of a person is deemed to be mandatory. The template used for displaying person entries can be augmented with a query to test for the existence of the `full name` field. A custom notification template can be shown if the query does not find a `full name` field. An example of this construction is illustrated in Figure 3.4.

As an added benefit of using the wiki system for describing custom notifications, it is possible for users to annotate the notifications. This can be used to add practical notes and references to guidelines or manuals to help out other users of the system

72 Consistency checks through describing structure As stated earlier, Strata does not impose a specific structure on the data entered by users. Any key can be used, and no special meaning is attached to those keys. However, policies and regulations typically include prescriptions on the structure of experimental and organisational data. Consistency checks can be used to gain insight into the current level of compliance with regulations.

Instead of having a separated schema definition language or system, the exact same system used to enter experimental and organisational data can be used to enter the expected schema. In doing so, all features that are available in Strata also apply to the schema: notes can be placed with the schema, a record of changes in the schema is kept, and templates can be used to present

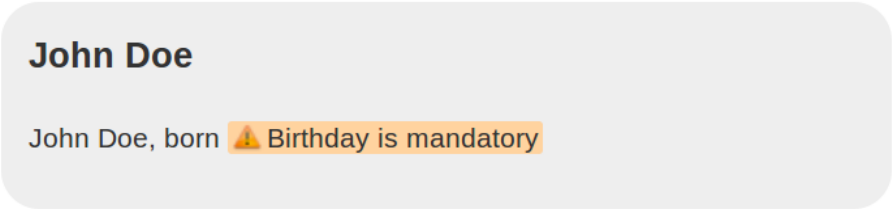
```
<template>
<block round box 50%>

//**@@entry title@@**//

@@Full Name@@, born
<*if Birthday> @@Birthday[date]@@ </if>
<*if !Birthday>
    <inline round important> Birthday is mandatory </inline>
</if>

</block>
</template>
```

(a) Template definition for person class, with notification for missing `Birthday` field.



(b) Result of template application to data entry with missing `Birthday` field.

Figure 3.4: Strata template with notification for missing `Birthday` field.

the schema in a clean and intuitive manner.

With the schema present as data, consistency checks boil down to a set of queries constructed to check the instances of the schema against the description of the schema. A consistency query matches all instances that do not match the schema, and templates can be used to combine the results with an explanation that relates to policy.

By describing the schema itself in the same system as the data, it becomes possible to document and annotate the schema, and to collaborate on the improvement of the schema.

3.5 Validation

The validation of the Strata system has been performed, under our supervision, by Daniel Davison. This section summarises the use case and his master's thesis [25] on designing a proof-of-concept automated lab notebook specialised for the workflow of biomedical research laboratories, using Prometheus laboratory of the KU Leuven as the investigated case.

73 Tissue engineering at Prometheus Prometheus, the skeletal tissue engineering department at the KU Leuven, works together with the Bone4Kids foundation. The institute hopes to develop an alternative treatment method for children suffering from pseudoarthrosis.

Pseudoarthrosis is a condition where a sustained bone fracture is unable to heal, resulting in a permanently broken bone. Current treatments often involve numerous complicated operations, prolonged revalidation or, in the worst case, can result in an amputation of the affected limb. By using skeletal tissue engineering techniques, researchers are developing a treatment which can replace the fractured bone segment with a healthy transplant grown from the patient's own donor cells. Success with this method should significantly reduce the amount of surgery and revalidation necessary, increasing the quality of the patient's life.

The researchers in the laboratory are associated with various tracks. Each research track focuses on a specific stage in the process of bone formation, ranging from fundamental research to computational modelling.

A large part of the data workflow in the laboratory relies heavily on standardised processes called Standard Operating Procedures. An SOP is defined by the International Conference on Harmonisation [48] as "detailed, written instructions to achieve uniformity of the performance of a specific function". In practice, this is typically a detailed a step-by-step instruction of the process, including details on the used materials, equipment and methods. A collection of such SOPs dictate the workflow of a research project. Since laboratory technicians are expected to operate according to the applicable

SOPs, work performed by different researchers can be more easily compared and shared.

74 Bottlenecks in the current paper-based workflow Based on an analysis of the current workflow at the Prometheus laboratory, the following major bottlenecks of the process have been traced back to the paper lab notebook:

- Archived information is relatively unstructured and difficult to retrieve.
- Collaboration is hindered, especially when considering remote partnerships.
- Organisational data (such as SOPs, cell lines, and patient and animal records) are difficult to establish and regulate.
- There is no possibility for automating tasks, such as retrieving measurements from a machine, or grouping all data for an experiment spread over time.

Retrieving information from archives requires manually browsing through the many notebooks in the archives and copying the necessary information. For example, to investigate the developments of a certain procedure over time, this will typically involve manually browsing through countless notebooks, copying the required information by hand, with the risk of introducing errors in the data.

Since the paper notebooks are the only authoritative source of data, it is very difficult to collaborate with others not present in the lab. Even within the laboratory it is difficult to ascertain the provenance information of a specific cell line. This information is spread across several notebooks, and researchers often do not update the shared spreadsheet document.

Further, not every type of record is standardised. For example, it was discovered that there is no standardisation for storing animal records, leading to several unique approaches to noting down this information. It can be argued

that this is a regulatory issue, but the presence of a digital system would greatly aid in going through with any standardisation effort.

Taking measurements often involves painstakingly copying several dozen numbers or tables by hand from a computer screen or paper print into the notebook, increasing the chance that errors are made in the process. Since a researcher is typically working on various experiments simultaneously, but only has one book, results from different experiments are organised criss-cross throughout the notebook. The researcher must keep close track of which pages belong to which experiment, and must manually introduce "jumps" from one page to another when there is not sufficient room to record all data on a single page.

75 Prototyping the automated lab notebook The prototype system implements a data schema that models organisational data. The data schema is modelled with a class-based approach: the concepts as present in the workflow (SOPs, cell lines, patient and animal records, etc.) are each modelled as a distinct class of data that features certain fields and constraints.

Referential constraints are defined as well, describing what classes can be referred to for fields that refer to other data. This greatly improves both the expressiveness of the schema, and the level of information that can be presented to the user in notifications and warnings.

The Strata syntax is relatively straightforward and can be taught to new users without much issue. Nevertheless, creating valid data blocks can become a daunting task when data models grow larger and more complex. Strata syntax is fully text-based and offers no real-time help when entering data. A user is presumed to know exactly how all data fields should be entered.

The solution used in the prototype system is the automatic creation of user friendly forms based on the extensive information about field names and relations in the schema. The forms offer automatic completion based on referential constraints, and assist the user in adding dates, and several kinds of measurements.

Tangentially related, the prototype system also includes rudimentary sup-

port for quantifiable data such as measurements that have been made during experiments. In order to reason about the experiment data, it is essential that one can query and compare entered values such as measurements and other stored quantities. Such measurements typically represent a physical quantity and are thus stored with the corresponding unit. The prototype extends Strata to handle physical quantities and units in a similar manner to the methods used by the Semantic WikiPedia extension [68].

76 Results at Prometheus The current prototype focuses primarily on matching the flexibility of the paper lab notebook by using free-form data principles while offering significant advantages in collaboration and data searchability:

- Firstly, searchability of research data is increased due to the indexation of the unstructured content in the wiki and the formalisation of the structured content in the semantic database.

- Secondly, use of a wiki platform as the basis for the lab notebook facilitates collaboration between researchers [59].

- Finally, the use of data models makes it possible to disclose the intended structure of semantic data, increasing the overall clarity, usability and maintainability of the system.

Although the proof of concept developed for this research illustrates how the data model approach can be applied to the workflow of a research laboratory, it is still somewhat limited in how it can be used in a real-world scenario.

The current interface was developed with a small scale proof of concept in mind, and several user interaction constructs that were used do not scale well when the wiki grows larger. An example of this is the relatively straightforward approach that is used when entering data: a dropdown shows the possible classes that are available in the system. This list will grow significantly larger as the wiki matures in a real-world setup, so an alternative approach is recommended.

A second major limitation of the current prototype is related to legal regulations related to electronic lab notebooks. Especially when clinical research is involved, a lab notebook is required to adhere to strict guidelines concerning the availability of data provenance, access restrictions and data integrity checks. Mitigating factors can be brought to the system in the form of access control and digital signing, but these have not been implemented in the proof of concept.

77 Other systems Next to the validation at Prometheus, the Strata system has been in use for a significant time in two other capacities.

The ASAS student project tracking system of the Database research group at the University of Twente has been constructed with Strata. In the workflow this system supports, the same tensions can be identified. Supervisors want to quickly update notes and the status of a project, and they want to do so in their own way. The interests of the research group as a whole are to have a clear and correct overview of the progress of master projects, and the distribution of work over the different supervisors. At the time of writing, the ASAS system has been in use for over three years by members of the research group.

Figure 3.5 shows the front page of the ASAS system. The leftmost navigation pane features links to the consistency, template and schema pages, as well as manuals and how-to's written be the users of the system. The lists of supervisors and current students are automatically created from the data in the system, as is the work load pie chart on the front page.

The online game FWURG[1] uses the Strata system for the representation and manipulation of game data. FWURG is a game played in turns, with each turn lasting a week. At the end of the week, the orders submitted by each player are checked and approved. Here too, the same structure of tension can be identified: players want to update their planets and trade routes in their own way, experimenting with different configurations of their economy and trades. The interests of the game as a whole is for the game data to adhere to certain rules and constraints such that the workflow of checking and approving

[1] `http://www.fwurg.net`

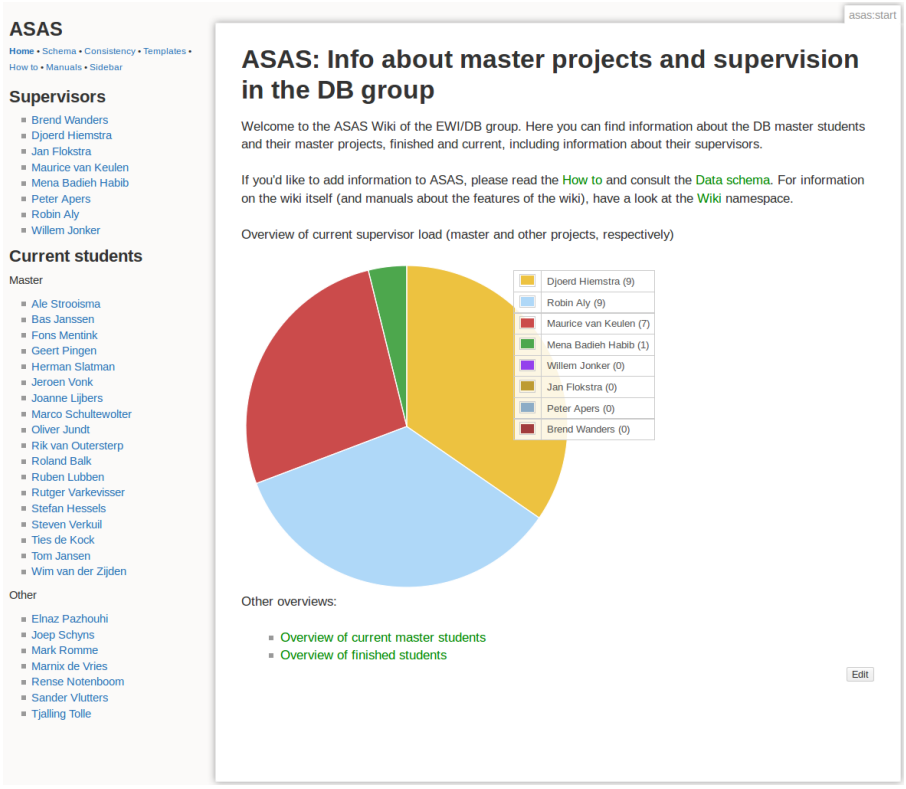


Figure 3.5: The ASAS system front page.

the player's orders proceeds smoothly. At the time of writing, the FWURG game has existed for five years, with a diverse group of players many of which do not have a technical background.

Figure 3.6 shows the detail page of the planet Mirda on the FWURG site. The figure shows the information box of the planet itself, which is constructed from several templates based on the available data. The underlying Strata system is also responsible for the navigation bar for the Smi-Halek System on the left of the screen, as well as the in-universe advertisement in the top right.

Figure 3.6: Detail page for the planet Mirda in the FWURG game.

3.6 Conclusions

In this chapter we have looked at the practice of note taking through the lens of a traditional research laboratory. We highlighted the opposing desires of the scientist and the institute, and have investigated the compromise that has been established in such environments.

Based on this compromise we sketched the three key features of freeform note taking, eventual observation of policy, and continuous policy evaluation that form the basis of our approach to automated support for the well-established compromise.

We presented the Strata system that implements the building blocks necessary to construct an automated lab notebook. The abilities of the system

have been validated by prototyping a lab notebook system for the Prometheus laboratory.

The lens of the traditional laboratory has provided us with a way to investigate the larger issue of note taking, by highlighting the problems that are associated with it and showing how those issues have been coped with in well-established fields of science.

While fourth paradigms institutes are well underway, through the review of ethics boards when handling personal or sensitive data, we can learn from the traditional laboratories. Looking at the achieved level of scientific rigour and accountability, we are of the opinion that fourth paradigm institutes should adopt Standard Operating Procedures for data integration and integrity.

CHAPTER 4

Framework for Probabilistic Databases

Parts of this chapter have been published as [114].

After the previous chapter's discussion of the value of documentation and freedom in data integration, we now focus on the technical aspects of our work. The method as described in Chapter 2 integrates data sources by viewing them as uncertain truths. This chapter is the start of the investigation of uncertainty for the purpose of data integration.

78 Dealing with uncertainty One of the core problems in soft computing is dealing with uncertainty in data. For example, many data activities such as data cleaning, coupling, fusion, mapping, transformation, information extraction, etc. are about dealing with the problem of semantic uncertainty [60, 75]. In the last decade, there has been much attention in the database community to scalable manipulation of uncertain data.

Probabilistic database research produced numerous uncertainty models and research prototypes, mostly relational. Examples of such systems are Trio [83], MayBMS [6, 56], MCDB [58] all aiming to represent uncertainty in the data they store and query. We refer to [87, Chapter 3] for an extensive survey of uncertain relational databases and their extensions. Other research focuses on XML as the data model of choice, this work is surveyed and discussed in the excellent survey paper by Abiteboul et al. [2]. There is work on probabilistic RDF, e.g., uRDF [77] and work by Rienstra [94]. Yet another direction are

probabilistic logics such as pD [37] and ProbLog [92], to which we will return in Chapter 5.

79 Motivation A growing number of approaches for soft computing data processing tasks are based on the application of probabilistic database technology. Some examples of these tasks are indeterministic deduplication [88], probabilistic XML data integration [62], and probabilistic integration of data about groupings as discussed in Chapter 6.

In these approaches, the data models vary while the uncertainty models seem highly similar. This raises the question if it is possible to obtain more uniformity, i.e., to define a uniform uncertainty model that can be applied to the various data models.

Furthermore there is an ongoing discussion at various workshops and conferences that is less visible in papers. This recurring discussion is about the use of different uncertainty models based on probability theory, fuzzy set theory, or Dempster-Shafer theory. This discussion raises the question of how these uncertainty models relate to uncertain data.

Based on these experiences, we find that there are still important open problems in dealing with uncertain data and that the available systems are inadequate on certain aspects. We address the following four aspects.

80 Insufficient understanding of core concepts Uncertainty in data has been the subject of research in several research communities for decades. Nevertheless, we believe our understanding of certain concepts is not deep enough. For example, *truth* of facts that are uncertain. Or, what are *possible worlds* really? Also, many models support possible alternatives in some way often associated with a probability. Are these *probabilities* loose add-ons or are they tightly connected to the alternatives?

81 Data model independence Depending on the requirements and domain, we use different data models such as relational, XML, and RDF. The available models for uncertain data are tightly connected to a particular data

model, resulting in the non-uniform handling of uncertainty in data as well as replication of functionality in the various prototype systems.

82 Aggregates In many data processing tasks, being able to aggregate data in multiple ways is essential. Computing aggregates over uncertain data is inherently exponential. There is much work available on approximating aggregates, often with error bounds, but this does not seem to suffice in all cases. Furthermore, systems offer operations on uncertain data as aggregates, such as EXP (expected value) in Trio [83]; they seem different from traditional aggregates such as SUM (summation), or is there a more generic concept of aggregation that encompasses all?

83 Optimisation opportunities There has been some work on optimisation for probabilistic databases, for example, in the context of MayBMS [56] and SPROUT [86], but as we experienced in [116] and Chapter 6, where we apply MayBMS to a bioinformatics homology use case, the research prototypes do not scale well enough to thousands of random variables. By generalising certain concepts in our formal foundation, we hope to create better understanding of optimisation opportunities.

84 Outlook We address the above with a new formalisation of a probabilistic database and associated notions as a result of revisiting its fundaments. The formalisation has the following properties:

- Data model independence,
- meta-data about uncertain data loosely coupled to raw data,
- loosely coupled probabilities, and a
- unified view on *aggregates* and *probabilistic database-specific functions*.

We demonstrate the usefulness of the formalisation for creating more insight by discussing questions like “What *are* possible worlds?”, “What is truth in an uncertain context?”, “What are aggregates?”, and “What optimisation

opportunities come to light?". Furthermore, we present a formal framework that can be applied to a certain data model to create a probabilistic variant.

We follow up with Chapter 5, where we validate data model independence by showing how to a obtain a probabilistic variants of Datalog, XPath, and relational algebra by applying the framework to their non-probabilistic counterparts. Finally, we finish our investigation in Chapter 6 with a validation of the real-world applicability of the framework to the problem of integrating homology grouping data.

4.1 Formal Framework

The basis of our formalism is the possible world. We use the term possible world in the following sense: as long as the winning number has not been drawn yet in a lottery, you do not known the winner, but you can envision a possible world for each outcome. Analogously, one can envision multiple possible database states depending on whether certain facts are true or not.

For example in Figure 1.2, a possible world (the true one) could contain annotations 1, 7, 8, and 10, but to a computer a world with annotations 2, 4, 5, and 6 could very well be possible too. Note that this differs from the use of the term 'possible world' in logics where it means possible interpretations [16, Chp.6] or as in modal logics [21].

The core of our formalisation is the idea that we need to be able to *identify* the different possible worlds so we can reason about them. We do this by crafting a way to incrementally and constructively describe the *name* of a possible world.

4.1.1 Representation

Our formalisation begins with the notion of a database as a possible world. A *database* $DB \in \mathbb{P}\, A$ consists of assertions $\{a_0, a_1, \ldots, a_n\}$ with a_i taken from A, the universe of assertions. For the purpose of data model independence, we abstract from what an assertion is: it may be a tuple in a relational database, a

node in an XML database, and so on. Since databases represent possible worlds, we use the symbols DB and w interchangeably. A probabilistic database PDB is a set of databases $\{DB_0, DB_1, \ldots, DB_n\}$, i.e., $PDB \in \mathbb{P}\,\mathbb{P}\,A$. Each different database represents a possible world in the probabilistic database. In other words, if an uncertainty is not distinguishable in the database state, i.e., if two databases are the same, then we regard this as one possible world. When we talk about possible worlds, we intend this to mean 'all possible worlds contained in the probabilistic database' denoted with W_{PDB}.

85 Implicit possible worlds Viewing it the other way around, an assertion holds in a subset of all possible worlds. To describe this relationship, we need an identification mechanism to refer to a subset of the possible worlds. For this purpose, we introduce the method of partitioning. A *partitioning* ω^n splits a database into n disjunctive parts each denoted with a *label* l of the form $\omega{=}v$ with $v \in 1..n$. If a world w is labelled with label l, we say that "l holds for w". Every introduced partitioning ω^n is a member of Ω, the set of introduced partitionings. W_l denotes the set of possible worlds in PDB labelled with l. $L(\omega^n) = \{\omega{=}v \mid v \in 1..n\}$ is the set of labels for partitioning ω^n.

In essence, possible worlds are about choices: choosing which assertions are in and which assertions are out. Independent choices may be composed, i.e., with k partitionings ω^n we obtain in the worst case n^k possible worlds.

86 Descriptive assertions and sentences A *descriptive assertion* is a tuple $\hat{a} = \langle a, \varphi \rangle$ where φ is a *descriptive sentence*, a propositional formula describing how the assertion relates to the possible worlds where the partitioning labels of the form $\omega{=}v$ are the only type of atomic sentences. $\top$ denotes the empty sentence logically equivalent with *true* and $\bot$ the inconsistent sentence logically equivalent with *false*. The usual equivalence of sentences by equivalence of proposition formulae applies with the addition that $v_1 \neq v_2 \implies \omega{=}v_1 \wedge \omega{=}v_2 \equiv \bot$. Note that these descriptive sentences are a generalised form of the world set descriptors of MayBMS [6]. The functions $a(\hat{a})$ and $\varphi(\hat{a})$ denote the assertion and sentence component of tuple $\hat{a}$, respectively. The

evaluation function $W(\varphi)$ determines the set of possible worlds for which the sentence holds. It is inductively defined as:

$$W(\omega{=}v) = W_{\omega=v} \tag{4.1}$$
$$W(\varphi \vee \psi) = W(\varphi) \cup W(\psi) \tag{4.2}$$
$$W(\varphi \wedge \psi) = W(\varphi) \cap W(\psi) \tag{4.3}$$
$$W(\neg\varphi) = W_{PDB} - W(\varphi) \tag{4.4}$$
$$W(\top) = W_{PDB} \tag{4.5}$$
$$W(\bot) = \varnothing \tag{4.6}$$

87 Compact probabilistic database A *compact probabilistic database* is defined as a set of descriptive assertions and a set of partitionings: $CPDB = (D, \Omega)$. We consider *CPDB well-formed* iff all labels used in D are a member of Ω and all assertions are present only once hence with one descriptive sentence: $\forall \hat{a}_1, \hat{a}_2 \in D : \hat{a}_1 \neq \hat{a}_2 \implies a(\hat{a}_1) \neq a(\hat{a}_2)$. A non-well-formed compact probabilistic database can be made well-formed by reconstructing Ω from the labels used in D and merging the 'duplicate' tuples using the following transformation rule:

$$\langle a, \varphi\rangle, \langle a, \psi\rangle \mapsto \langle a, \varphi \vee \psi\rangle \tag{4.7}$$

88 Naming a possible world In general, φ denotes a set of possible worlds. The most restrictive set of worlds is described by a *fully described sentence* $\bar{\varphi}$ constructed as a conjunction of labels for each introduced partitioning of Ω. Because of well-formedness and because a possible world is only distinguished by the assertions it consists of, it follows that $\bar{\varphi}$ describes a single possible world. For example, given that $\Omega = \{x^2, y^3, z^2\}$, one of the possible worlds is fully described by $x{=}1 \wedge y{=}2 \wedge z{=}2$.

Let $L(\Omega)$ be the set of all possible fully described sentences:

$$L(\Omega) = \{l_1 \wedge \ldots \wedge l_k \mid \Omega = \{\omega_1^{n_1}, \ldots, \omega_k^{n_k}\} \wedge \forall i \in 1..k : l_i \in L(\omega_i^{n_i})\} \tag{4.8}$$

The set of possible worlds contained in *CPDB* can now be defined as:

$$W_{PDB} = \bigcup_{\varphi \in L(\Omega)} W(\varphi) \tag{4.9}$$

Note that because each ω^n is a partitioning, the following holds:

$$\forall \omega^n \in \Omega : W_{PDB} = \bigcup_{l \in L(\omega^n)} W_l \tag{4.10}$$

89 Dependencies between assertions Dependencies in the existence between assertions can be expressed with descriptive sentences logically combining different labels.

- *Mutual dependency* can be expressed by using the same sentence for the tuples. For example, $\langle a, \varphi \rangle$ and $\langle b, \varphi \rangle$ describes the situation where a and b both exist in a possible world or neither, but never only one of the two.
- *Implication* can be expressed by containment. For example, $\langle a, \varphi \rangle$ and $\langle b, \varphi \wedge \psi \rangle$ describes the situation that whenever a is contained in a possible world, then b is too.
- *Mutual exclusivity* can be expressed with mutually exclusive sentences, i.e., $\langle a, \varphi \rangle$ and $\langle b, \psi \rangle$ can never occur together in a possible world if $\varphi \wedge \psi \equiv \bot$.

Since each ω is a partitioning on its own, they can be considered as *independent* choices. For example, $\langle a, x{=}1 \rangle$ and $\langle b, y{=}1 \rangle$ use different partitionings, hence the labels establish no dependency between a and b and thus the existence of a and b is independent.

4.1.2 Querying

The concept of possible worlds means that querying a probabilistic database should be indistinguishable from querying each possible world separately, i.e., producing the same answers.

90 Multiple databases, multiple answers Querying a database produces an answer based on the given query. Querying multiple databases with the same query produces multiple answers, one for each database. This principle is extended to querying over a compact probabilistic database. Since the compact probabilistic database represents multiple databases, it produces multiple answers:

$$\begin{array}{ccccc} \{DB\} & \xleftrightarrow{f} & PDB & \xleftrightarrow{c} & CPDB \\ \downarrow{\scriptstyle \oplus} & & \downarrow{\scriptstyle \oplus} & & \downarrow{\scriptstyle \hat{\oplus}} \\ \{R\} & \xleftrightarrow{f} & PR & \xleftrightarrow{c} & CPR \end{array} \tag{4.11}$$

Equation 4.11 illustrates the relationships between a set of databases, a probabilistic database, a compact probabilistic database and the associated query results. The operations f and c represent formation and compaction, respectively. Formation constructs a probabilistic database from a set of databases. Compaction takes a probabilistic database and produces a compact probabilistic database. Both operations are trivially inverted as f' and c', through unpacking and enumerating all possible worlds, respectively. $\oplus$ is the normal query operator, and $\hat{\oplus}$ is the query operator extended to apply to a compact probabilistic database. This extension is discussed in the next paragraph.

91 Extending the query operator For any query operator $\oplus$, we define an *extended operator* $\hat{\oplus}$ with an analogous meaning that operates on a compact representation. It is defined by $\hat{\oplus} = (\oplus, \tau_{\oplus})$. Where $\tau_{\oplus}$ is a function that produces the descriptive sentence of a result based on the descriptive sentences of the operands, and in a manner that is appropriate for operation $\oplus$. We call an extended operator sound iff it adheres to the commutative relations of Equation 4.11. This means, for example, that $\hat{\oplus} = (c \circ \oplus \circ c')$. Alternatively, starting from the non-compact probabilistic database PDB, the equality $(c \circ \oplus) = (\hat{\oplus} \circ c)$ must hold for any $\hat{\oplus}$.

Observe that we abstract from specific operators analogously to the way we abstract from the form of the actual data items. The above defines how to

construct probabilistic operators from non-probabilistic ones. In this way, one can apply this to any query language in effect defining a family of probabilistic query languages.

4.1.3 Probability calculation

One can attach a probability $P(\omega{=}v)$ to each partition v of a partitioning ω^n provided that $\sum_{v=1}^{n} P(\omega{=}v) = 1$. As is known from the U-relations model [6] and variations thereof such as [60], calculating probabilities of possible worlds or the existence of an assertion among the worlds, can make use of certain properties that also apply here. For example, $P(\omega_1{=}v_1 \wedge \omega_2{=}v_2) = P(\omega_1{=}v_1) \times P(\omega_2{=}v_2)$ and $P(\omega_1{=}v_1 \vee \omega_2{=}v_2) = P(\omega_1{=}v_1) + P(\omega_2{=}v_2)$ iff $\omega_1 \neq \omega_2$. Moreover,

$$P(\langle a, \varphi \rangle) = \sum_{\substack{w \in W_{PDB} \\ a \in w}} P(w) \tag{4.12}$$

$$= \sum_{w \in W(\varphi)} P(w) \tag{4.13}$$

$$= P(\varphi) \tag{4.14}$$

Constraining the expressiveness of the descriptive sentences or requiring a normal form may allow for more time-efficient exact probability calculations beyond enumerating all worlds, for example, [66] describes an tractable approach for calculating the probabilities of positive sentences in disjunctive normal form. Larger amounts of uncertainty, represented by large amounts of partitionings involved in the description of a possible world, may require approximate probability calculation to remain feasible. One such approach to this problem is detailed in [85].

4.2 Example: Fruit Salad

Before we discuss the framework in the following sections, we want to show a minimal example of how the framework can be used. To do so, we present the following example.

92 Knowledge, wisdom and fruit salads When preparing a fruit salad, most of us keep in mind that *"Knowledge is knowing that a tomato is a fruit, wisdom is not putting it in a fruit salad."*

Before we investigate the ramifications of this statement, we sketch the mathematical context of this example. We posit the sets *Fruits* and *Vegetables* that contain all fruits and all vegetables, respectively, and where $\mathit{Fruits} \cap \mathit{Vegetables} = \varnothing$. We limit our universe of discourse to those edibles $e \in \mathit{Edibles}$ for which $e \in \mathit{Fruits} \cup \mathit{Vegetables}$ holds. Further, we posit that when a chef picks ingredients for a fruit salad he is constrained by $\forall e \in \mathit{Salad} : e \in \mathit{Fruits}$ leading to $\mathit{Salad} \subseteq \mathit{Fruits}$.

93 Doubt about the chef We know that for any edible $e \in \mathit{Edibles}$ either $e \in \mathit{Fruits}$ or $e \in \mathit{Vegetables}$ is true. For some edibles, such as *apple*, it is eminently clear into which of the two subsets they fall. For *tomato* there is less clarity. The tomato might be in either of the two sets, i.e., $\mathit{tomato} \in \mathit{Fruits}$ or $\mathit{tomato} \in \mathit{Vegetables}$, depending on one's knowledge.

There is an element of uncertainty here. Is the chef knowledgeable? If so, is he wise? And what ingredients will he pick for the salad? To model the impact of this uncertainty, we want to create a probabilistic variant of the Set Membership data model.

We briefly define the data model and query operations, this definition is followed by an application of the framework to the data model. The result is a probabilistic variant of the model.

4.2.1 Framework applied to Set Membership

The 'Set Membership' data model is a simple data model based on determinations. Each determination states the membership of an element in a set. How this determination is made is irrelevant for the data model. The 'Set Membership' model can be queried for a specific set, the answer being all elements that are a member of the set.

To clearly denote a 'Set Membership' determination as different from the normally used set membership symbol, it is denoted as the symbol δ, followed by a parenthesised element and set. For example, the determination $\delta(\mathit{apple}, \mathit{Fruits})$ asserts that $\mathit{apple} \in \mathit{Fruits}$.

94 Definition of Set Membership We postulate disjoint sets *Elem* and *Set* as the sets of *elements* and *sets* respectively. Let $e \in \mathit{Elem}$ and $S \in \mathit{Set}$. A determination $d = \delta(e, S)$, consisting of an element and a set, represents that the element e is a member of set S, i.e., $e \in S$. A set of determinations $\mathrm{KB} = \{d_1, \ldots, d_n\}$ is called a *knowledge base*.

An example of a set membership knowledge base, based on our fruit salad example and a chef that is not knowledgable, is as follows:

$$\begin{array}{rr} \delta(\mathit{apple}, \mathit{Fruits}) & \delta(\mathit{tomato}, \mathit{Vegetables}) \\ \delta(\mathit{banana}, \mathit{Fruits}) & \delta(\mathit{carrot}, \mathit{Vegetables}) \\ \delta(\mathit{mango}, \mathit{Fruits}) & \delta(\mathit{broccoli}, \mathit{Vegetables}) \end{array} \tag{4.15}$$

This example contains determinations to state that $\mathit{Fruits} = \{\mathit{apple}, \mathit{banana}, \mathit{mango}\}$ and $\mathit{Vegetables} = \{\mathit{tomato}, \mathit{carrot}, \mathit{broccoli}\}$.

A set membership knowledge base can be queried with the $\Sigma(S)$ query operation. The answer to such a query is a set of all elements that have been determined to be a member of S in the knowledge base. In other words, the result of the query $\Sigma(S) = \{e \mid \delta(e, S) \in \mathrm{KB}\}$. For example, based on our previous sample determinations, the result of the query $\Sigma(\mathit{Fruits})$ is $\{\mathit{apple}, \mathit{banana}, \mathit{mango}\}$.

95 Probabilistic Set Membership To obtain a probabilistic set membership model using our framework we view the determinations as assertions. We use the notation $\delta(e,S)^{[\varphi]}$ for the descriptive tuple $\hat{d} = \langle \delta(e,S), \varphi \rangle$. In this way, we can specify uncertain determinations, as well as dependencies between determinations. Descriptive sentences where $\varphi \equiv \top$ may be omitted for brevity.

The operation from our framework is Σ, and applying our framework means defining $\hat{\Sigma}$ by defining τ_Σ and weaving it into the given definition of Σ:

$$\varphi \overset{\tau_\Sigma}{\mapsto} \varphi$$

$$\frac{d \in \mathrm{KB} \quad d = \delta(e,S)}{e \in \Sigma(S)} \quad \overset{\Sigma}{\longrightarrow} \quad \frac{\hat{d} \in \mathrm{KB} \quad \hat{d} = \delta(e,S)^{[\varphi]} \quad \varphi' = \tau_\Sigma(\varphi) \quad \varphi' \not\equiv \bot}{\langle e, \varphi' \rangle \in \hat{\Sigma}(S)} \tag{4.16}$$

The intuition behind this definition is that an element e is only a member of set S if its descriptive sentence φ holds. Note that the result of $\hat{\Sigma}$ is a compact representation of multiple sets where each potential member of the set S has an attached descriptive sentence.

4.2.2 The possible worlds

Armed with a probabilistic version of the set membership model, we can now start to represent the uncertainty about the chef's knowledge and wisdom.

Following the structure of *"Knowledge is knowing that a tomato is a fruit, wisdom is not putting it in a fruit salad"*, we first envision only two worlds. One where the chef is knowledgeable, and where where he is not.

96 Knowledge and the tomato We can represent these two worlds extending the example above by introducing some uncertainty about the status of the *tomato*. To name these possible worlds, we introduce the partitioning k^2, with labels k=1 and k=0.

Now, we have two worlds. The first, called k=1 expresses a world with a knowledgeable chef: $\mathit{Fruits} = \{\mathit{apple}, \mathit{banana}, \mathit{mango}, \mathit{tomato}\}$. The second,

called k=0 expresses a world with an unknowing chef: $\mathit{Fruits} = \{\mathit{apple}, \mathit{banana}, \mathit{mango}\}$. The two possible worlds differ only in the chef's knowledge (note that we have added determinations for the *Salad*):

$$\begin{array}{ll} \delta(\mathit{apple}, \mathit{Fruits})^{[\top]} & \delta(\mathit{carrot}, \mathit{Vegetables})^{[\top]} \\ \delta(\mathit{banana}, \mathit{Fruits})^{[\top]} & \delta(\mathit{broccoli}, \mathit{Vegetables})^{[\top]} \\ \delta(\mathit{mango}, \mathit{Fruits})^{[\top]} & \delta(\mathit{banana}, \mathit{Salad})^{[\top]} \\ \delta(\mathit{tomato}, \mathit{Fruits})^{[k=1]} & \delta(\mathit{mango}, \mathit{Salad})^{[\top]} \\ \delta(\mathit{tomato}, \mathit{Vegetables})^{[k=0]} & \delta(\mathit{tomato}, \mathit{Salad})^{[k=1]} \end{array} \tag{4.17}$$

If we query the knowledge base for $\hat{\Sigma}(\mathit{Salad})$ we find that the result comes with *tomato* included only for worlds where k=1:

$$\hat{\Sigma}(\mathit{Salad}) = \{\langle \mathit{banana}, \top\rangle, \langle \mathit{mango}, \top\rangle, \langle \mathit{tomato}, k{=}1\rangle\} \tag{4.18}$$

Note that $\delta(\mathit{tomato}, \mathit{Salad})$ has an attached descriptive sentence φ of k=1. This is because of the $\mathit{Salad} \subseteq \mathit{Fruits}$ constraint: the *tomato* can only be picked as an ingredient if the chef thinks it is a fruit.

97 Wisdom and the tomato Now, we introduce uncertainty about the wisdom of the chef. This requires the introduction of the partitioning w^2 to describe the "choice" between a wise and an unwise chef. The labels w=1 and w=0 indicate a wise and unwise chef, respectively.

We can update our knowledge base with this new partitioning, making sure not to use the *tomato* in the *Salad* if the chef is wise:

$$\begin{array}{ll} \delta(\mathit{apple}, \mathit{Fruits})^{[\top]} & \delta(\mathit{carrot}, \mathit{Vegetables})^{[\top]} \\ \delta(\mathit{banana}, \mathit{Fruits})^{[\top]} & \delta(\mathit{broccoli}, \mathit{Vegetables})^{[\top]} \\ \delta(\mathit{mango}, \mathit{Fruits})^{[\top]} & \delta(\mathit{banana}, \mathit{Salad})^{[\top]} \\ \delta(\mathit{tomato}, \mathit{Fruits})^{[k=1]} & \delta(\mathit{mango}, \mathit{Salad})^{[\top]} \\ \delta(\mathit{tomato}, \mathit{Vegetables})^{[k=0]} & \delta(\mathit{tomato}, \mathit{Salad})^{[k=1 \wedge w=0]} \end{array} \tag{4.19}$$

The knowledge base now describes four possible worlds. With the partitionings being $\Omega = \{k^2, w^2\}$, the fully described possible worlds are: $(k{=}0 \wedge w{=}0)$, $(k{=}1 \wedge w{=}0)$, $(k{=}1 \wedge w{=}1)$, and $(k{=}0 \wedge w{=}1)$.

The label $w{=}1$ is not used in any descriptive sentence in the knowledge base. Since no determination is needed to express the non-membership of an element, simply not having the $\delta(\mathit{tomato}, \mathit{Salad})$ determination in worlds with $w{=}1$ is enough.

Furthermore, by using a logical conjunction in $k{=}1 \wedge w{=}0$ we have introduced a dependency in this update: there will only be tomato's in the salad if the chef is knowledgeable but unwise. Possessing only a lack of wisdom is not enough.

4.3 Comparison with Possible Worlds

The above-described framework is in essence a generalisation of the U-relations model behind MayBMS [6]. Most other probabilistic database models [87, Chapter 3] are also based on the concept of possible worlds. Our framework mainly distinguishes itself from these models on the following aspects.

98 Independence from data model and uncertainty model We have abstracted from what the raw data looks like by treating it as assertions. In this way, we obtain data-model independence whereas other models are defined for a specific data model.

Furthermore, probabilities are separately attached as an 'optional add-on' obtaining the desired loose coupling between alternatives and probabilities.

Our formal foundation is a framework that turns a data model and query language into a probabilistic version, hence we have not defined one specific model but a family of models.

99 Separation of data and uncertainty metadata The descriptive sentences represent the uncertainty metadata. As it is nicely separate from the raw data, we obtain a loose coupling between data and uncertainty metadata. This allows the development of a generic uncertainty management component

that can be reused in systems using different data models. The uncertainty management functionality of existing prototypes is built into the probabilistic database itself and cannot easily be reused when developing another.

100 Dependencies We allow full propositional logic for constructing descriptive sentences which results in an expressive mechanism for establishing complex dependencies. As further discussed in Section 5.2.5, [2] showed for probabilistic XML that certain probabilistic XML families are fundamentally more expressive than the other families even while they only allow conjunctions of independent events whereas we allow any propositional sentence.

4.4 Discussion

In this section we discuss the implications of our framework. We also revisit the open problem of aggregates, and discuss optimisation opportunities.

101 Three kinds of (un)truth A probabilistic variant of a language like Datalog (as we define in Section 5.1) is an interesting vehicle to obtain deeper understanding of important concepts such as *truth* of facts that are uncertain. In fact, the language can express three kinds of untruth.

1. A fact A is entailed with an inconsistent sentence $\varphi \equiv \bot$. This means that although A seems logically derivable, its derivation implies that the world is impossible, i.e., it is true in none of the possible worlds.

2. A fact A is entailed with a sentence φ with $P(\varphi) = 0$. This means that A is derived only for worlds with zero probability.

3. A fact A is not entailed (in any of the possible worlds). This is the original untruth of Datalog.

The differences between these untruths are rather subtle but they exist.

4.4.1 Optimisations

102 Scalable uncertainty A probabilistic database not only needs to be scalable in the volume of data, but also in the amount of uncertainty in the data. The latter presents itself both in the number of partitionings as well as in the size of the descriptive sentences. From our experience with a bioinformatics use case [116] and Chapter 6, the number of partitionings can easily grow into the thousands in real-world applications. The size of the descriptive sentences is determined by the complexity of the dependencies between assertions, its low-level representation, and allowed expressiveness.

103 Propositional logic techniques As propositional logic is the basis of the descriptive sentence, many algorithmic techniques can be applied. Equivalence-based sentence rewriting can be used for, simplification, normalisation, and negation removal (a negated label can be substituted with an exhaustive disjunction of the other labels in the partitioning). An example of optimisations based on disjunctive normal form is [66]. Negation removal is particularly useful if the partitionings are restricted to be binary, which may be sufficient for certain applications and allows for many other optimisations.

During query execution, assertions with an inconsistent sentence can be filtered out. This, as well as the sentence rewriting techniques, can be done eagerly or lazily depending on the trade-off between overhead of the technique and resulting gains. Sentence manipulation can be optimised by taking into account properties of the operations, e.g., selection is guaranteed to produce a well-formed unmodified result, so no rewriting or filtering is necessary.

On the implementation level, special physical operators can combine data processing with sentence manipulation. For example, a merge-join implementation of $\hat{\bowtie}$ could combine joining tuples with simplification, normalisation, and filtering.

104 Constraining expressiveness The full expressiveness of propositional logic allows for the expression of rich dependencies between assertions at the price of computational complexity. Restricting expressiveness can provide

optimisation benefits, e.g., disallowing negation may allow many optimisations that are not valid in its presence.

The data model and query language may already place lower requirements on the expressiveness of the descriptive sentences. For example, the only logical connective in probabilistic Datalog of Section 5.1 is conjunction, and disjunction is necessary for maintaining well-formedness. Hence, negation is not needed and also conjunction and disjunction only appear in particular patterns. Vice versa, restrictions on the descriptive sentences may restrict the query language as well. For example, without negation the difference between relations cannot be supported in probabilistic relational algebra.

105 Efficient probability calculation Calculation of exact probabilities for query results may be computationally expensive and even exceed processing of the query itself. This cost can be mitigated by (1) only calculating probabilities on-demand such as in Trio [83], (2) approximating probabilities given some error bound, (3) caching probability calculation results for long shared parts of frequently occurring descriptive sentences. Furthermore, applying simpler probabilistic models also allows for more efficient probability calculation, exact or approximate.

4.4.2 Open problems

106 Aggregates Equation 4.11 determines the semantics of traditional aggregates such as summation SUM in a probabilistic database:

$$\widehat{\text{SUM}} = (c' \circ \text{SUM} \circ c) \tag{4.20}$$

The difference with the other relational operators is that their direct computation over a compact probabilistic database is much less straightforward, because they may produce an answer that exponentially grows with growing

numbers of partitionings. For example, given a probabilistic relation:

$$R = \{\langle 1, x{=}1\rangle, \langle 2, x{=}1 \vee y{=}1\rangle, \langle 3, x{=}2 \wedge z{=}1\rangle, \langle 5, y{=}2\rangle\} \tag{4.21}$$

$$\Omega = \{x^2, y^2, z^2\} \tag{4.22}$$

the answer of $\widehat{\text{SUM}}(R)$ is:

$$\begin{aligned}\{&\langle 2, x{=}2 \wedge y{=}1 \wedge z{=}2\rangle,\\ &\langle 3, x{=}1 \wedge y{=}1\rangle,\\ &\langle 5, x{=}2 \wedge ((y{=}1 \wedge z{=}1) \vee (y{=}2 \wedge z{=}2)\rangle,\\ &\langle 8, y{=}2 \wedge (x{=}1 \vee (x{=}2 \wedge z{=}1))\rangle\}\end{aligned} \tag{4.23}$$

Observe that although not every possible world results in a different answer, it is an open problem how to construct sentences for the answers in an efficient way, i.e., without enumerating all worlds.

Note that in many applications it is not necessary to determine the full set of possible exact answers with their probabilities. The master's thesis of Knippers [65] proposes a variety of answer forms for aggregate queries that can be (more) efficiently computed and may still be sufficiently informative such as (a) a single value representing the expected value of the sum, (b) two values representing the mean of the sum and its standard deviation, (c) a histogram with probabilities for a predetermined number of answer ranges, (d) a single answer representing the single most likely value possibly with its probability, or (e) a top-k of the k most likely results, and so forth.

107 Out-of-world aggregations Many systems offer the expected value as an aggregation function. Furthermore, whereas computing a sum over probabilistic data has exponential complexity, computing the expected value of a sum has not. Therefore, such systems offer combined aggregators such as the 'esum'. This poses the questions of: are these truly aggregators?; and what is an aggregate really?

Traditional aggregates operate by aggregating values over a dimension,

possibly in groups, where a dimension typically is an attribute of a relation. The possible worlds can be seen as yet another dimension. For this reason, the expected value is indeed an aggregator, namely one operating over the possible worlds dimension. This insight has the potential of treating all aggregates, including the probabilisticly inspired ones, uniformly as well as combinations of aggregators. Note that asking for the probability of a tuple or for an expected value forms a new class of query operators: they have no counterpart in the non-probabilistic query language. More research is needed to explore the implications of this new class of queries.

4.5 Conclusions

We revisited the formal foundations of probabilistic databases by proposing a formal framework that is based on attaching a propositional logic sentence to data assertions to describe the possible worlds in which that assertion holds. By doing so, the formalisation (a) abstracts from the underlying data model obtaining data model independence, and (b) separates metadata on uncertainty and probabilities from the raw data.

In relation to the framework, we discuss open problems such as alternative data models, probability calculation, and aggregation, as well as scalability and optimisation issues brought to light due to the framework's properties.

Data model independence of the framework will be validated in Chapter 5 by applying it to Datalog, XPath and relational algebra to obtain probabilistic variants thereof: for every query operator $\oplus$, we define (a) sentence manipulation function $\tau_{\oplus}$ and (b) probabilistic query operator $\hat{\oplus}$, the latter by weaving $\tau_{\oplus}$ into the original definition of $\oplus$.

CHAPTER 5

Validation of Orthogonality

Parts of this chapter have been published as [114, 115].

We continue the presentation of our framework by validating the framework's data model independence. In this chapter we discuss the application of the framework on three different data models and associated query operations. In doing so we illustrate that the framework can be applied equally well to these different data models.

Each application proceeds in the same manner: we briefly define the data model and query operations, this definition is followed by an application of the framework to the data model. We discuss the resulting probabilistic variant of the model. We then present an implementation of this probabilistic variant, and continue by discussing possible optimisations.

The three data models that are investigated are Datalog, XML/XPath, and the relational model/SQL. We first give a brief introduction to all three.

108 Datalog: JudgeD Datalog is a query language for deductive databases. Datalog models both data and inference rules as restricted Horn clauses. The restrictions serve to guarantee termination of queries. In Section 5.1 we present JudgeD, our implementation of the probabilistic Datalog created by applying our framework to ordinary Datalog. Two separate solvers are presented: an exact implementation to determine sentences associated with answers, and a Monte Carlo approach to estimate the probabilities of answers. The JudgeD system has been used for data cleaning in maritime evidence combination [46].

109 XML/XPath XPath is a navigation-based query language for XML documents. In essence XML is a heterogeneous tree data model. Application of our framework extends this tree model to a probabilistic tree model. In Section 5.2 we present our implementation of probabilistic XPath based on a translation from probabilistic XPath to XQuery that can be executed on any XQuery engine.

110 Relational: MayBMS Relational algebra is the underpinning of relational databases. The data model features sets of tuples, with each set conforming to a specific schema that describes the structure of the tuples contained within it. Application of our framework on the relational model produces a probabilistic relational model much like the U-tables known from MayBMS. In Section 5.3, we present the application of our framework to the relational model. We show that MayBMS is an optimised implementation of a constrained variant. We continue this investigation in Chapter 6 where we use MayBMS to combine real-world grouping data from different sources with data on homologous relations between proteins.

5.1 Probabilistic Datalog: JudgeD

In this section we present JudgeD, a probabilistic Datalog in which probabilities can be attached to both factual data and rules. JudgeD has been motivated by our ongoing work on maritime evidence combination, where we want to reason with uncertain facts and rules expressing heuristics. The formal foundation of JudgeD is based on a direct application of the framework of Chapter 4 to Datalog. It therefore functions as the first of three data models with which we validate data-model orthogonality of our framework. We present a proof-of-concept implementation of both a Monte Carlo based answer probability approximation and an exact solver supported by Binary Decision Diagrams (BDDs).

Several probabilistic logics have been developed over the past three decades. Prominent examples include Probabilistic Horn Abduction [90]; PRISM

[97]; Stochastic Logic Programs (SLPs) [81]; Markov Logic Networks [93]; constraint logic programming for probabilistic knowledge, know as CLP($\mathcal{BN}$), [22]; probabilistic Datalog, known as pD, [37]; and ProbLog [92]. In these logics probabilities can be attached to logical formulas, under the imposition of various constraints. In SLPs clauses defining the same predicate are assumed to be mutually exclusive; PRISM and PHA only allow probabilities on factual data and under constraints that effectively enforce mutual exclusivity.

However, we observe that JudgeD — being the immediate result of applying our framework — can express complex dependencies between clauses. None of the above-mentioned probabilistic logics supports such expressiveness. In Section 5.1.3 we show that efficient reasoning is still possible by actually giving an implementation and presenting generic and Datalog-specific optimisation opportunities.

111 Contributions & Outlook

The key contributions in this section are:

- First part of the validation of the data model independence property of the framework of Chapter 4.
- The expression of dependencies between arbitrary clauses, both facts and rules (e.g., mutual exclusivity, independence, mutual dependence, implication and more complex dependency relations),
- The proof-of-concept implementation of both a Monte Carlo based approximation as well as an exact solver.

Section 5.1.1 describes JudgeD in terms of a direct application of our framework to Datalog. Section 5.1.2 presents the syntax and applicability of JudgeD based on the use case of Maritime Evidence Combination (see Section 1.7.2). Section 5.1.3 discusses the implementation of the system which which is based on two solvers: an exact implementation to determine sentences associated with answers, and a Monte Carlo approach to estimate the probabilities of answers. Finally, Section 5.1.4 discusses JudgeD in relation to other work,

Section 5.1.5 presents avenues for future work, and we conclude our discussion of JudgeD in Section 5.1.6.

5.1.1 Framework applied to Datalog

Datalog is a knowledge representation and query language based on a subset of Prolog. It allows the expression of facts and rules. Rules specify how more facts can be derived from other facts. A set of facts and rules is known as a Datalog program.

In the sequel, we first define our Datalog language and then apply the framework to obtain probabilistic Datalog by viewing the facts and rules as assertions. We base our definition of Datalog on [16, Chapter 6]. Although the presentation here concerns for simplicity positive Datalog only, the Monte Carlo-based solver supports a probabilistic variant of negative Datalog as well.

112 Definition of Datalog We postulate disjoint sets *Const*, *Var*, *Pred* as the sets of *constants*, *variables*, and *predicate symbols*, respectively. Let $c \in \mathit{Const}$, $X \in \mathit{Var}$, and $p \in \mathit{Pred}$. A *term* $t \in \mathit{Term}$ is either a constant or variable where $\mathit{Term} = \mathit{Const} \cup \mathit{Var}$.

An *atom* $A = p(t_1, \ldots, t_n)$ consists of an n-ary predicate symbol p and a list of argument terms t_i. An atom is *ground* iff $\forall i \in 1..n : t_i \in \mathit{Const}$. A *clause* or *rule* $r = (A^{\mathrm{h}} \leftarrow A_1, \ldots, A_m)$ is a horn clause representing the knowledge that A^{h} is true if all A_i are true. A *fact* is a rule without body $(A^{\mathrm{h}} \leftarrow)$. Let $\mathit{vars}(r)$ be the set of variables occurring in rule r. A set of rules KB is called a *knowledge base* or *program*. The usual safety conditions of pure Datalog apply.

An example of a Datalog program, based on our running example of natural language processing from Section 1.7.1, can be seen in Figure 5.1. It determines the country C of a phrase Ph at position Pos if it is of type place and refers to an entry in a gazetteer containing the country.

Let $\theta = \{X_1/t_1, \ldots, X_n/t_n\}$ be a *substitution* where X_i/t_i is called a *binding*. $A\theta$ and $r\theta$ denote the atom or rule obtained from replacing each X_i occurring in A or r by the corresponding term t_i.

```
type(paris, pos1, place) ←
gazetteer(g11, paris, france) ←
refersto(paris, pos1, g11) ←

location(Ph, Pos, C) ←
  type(Ph, Pos, place), refersto(Ph, Pos, G),
  gazetteer(G, Ph, C)
```

Figure 5.1: An example Datalog program based on the natural language example from Section 1.7.1.

Semantic entailment for our Datalog is defined in Figure 5.2 (left side of $\xrightarrow{\models}$) as the *Herbrand base*: all ground atoms that can be derived as a logical consequence from KB.

The three facts of our example are entailed directly, because their bodies are empty, hence $m = 0$, and the heads are already ground such that $\theta = \varnothing$ suffices. The location-rule contains variables. With $\theta = \{\mathsf{Ph}/\mathsf{paris}, \mathsf{Pos}/\mathsf{pos1}, \mathsf{G}/\mathsf{g11}, \mathsf{C}/\mathsf{france}\}$ or any superset thereof the atoms in the body turn into entailed facts allowing $\mathsf{location}(\mathsf{paris}, \mathsf{pos1}, \mathsf{france})$ to be entailed.

113 Probabilistic Datalog The approach to obtain Probabilistic Datalog using our framework is by viewing the facts and rules as assertions. We use the notation $(A^{\mathrm{h}} \overset{\varphi}{\leftarrow} A_1, \ldots, A_m)$ for the tuple $\langle A^{\mathrm{h}} \leftarrow A_1, \ldots, A_m, \varphi \rangle$. Note that viewing facts and rules as assertiongs not only allows the specification of uncertain facts, but also uncertain rules as well as dependencies between the existence of facts and rules. In this way, the Probabilistic Datalog we obtain is more expressive than existing flavours of probabilistic Datalog as mentioned in the introduction of Section 5.1.

The 'operation' in Datalog is entailment. Therefore, applying our framework means defining probabilistic entailment $\hat{\models}$ by defining $\tau_{\models}$ and weaving it into the given definition of $\models$ (see Figure 5.2). The intuition behind the definition is that the descriptive sentence of an entailed fact is the conjunction of the sentences of the atoms and rules it is based on, which should not be 'false', i.e., it should not be equivalent to the sentence $\bot$.

$$\varphi, \varphi_1, \ldots, \varphi_m \overset{\tau_{\models}}{\mapsto} \varphi \wedge \textstyle\bigwedge_{i \in 1..m} \varphi_i$$

$$\frac{\begin{array}{c} r \in \mathrm{KB} \quad r = (A^{\mathrm{h}} \leftarrow A_1, \ldots, A_m) \\ \exists \theta : A^{\mathrm{h}}\theta \text{ is ground} \wedge \forall i \in 1..m : \mathrm{KB} \models A_i\theta \end{array}}{\mathrm{KB} \models A^{\mathrm{h}}\theta} \quad \xrightarrow{\models}$$

$$\xrightarrow{\models} \quad \frac{\begin{array}{c} \hat{r} \in \mathrm{KB} \quad \hat{r} = (A^{\mathrm{h}} \overset{\varphi}{\leftarrow} A_1, \ldots, A_m) \\ \exists \theta : A^{\mathrm{h}}\theta \text{ is ground} \wedge \forall i \in 1..m : \mathrm{KB} \hat{\models} \langle A_i\theta, \varphi_i \rangle \\ \varphi' = \tau_{\models}(\varphi, \varphi_1, \ldots, \varphi_m) \quad \varphi' \not\equiv \bot \end{array}}{\mathrm{KB} \hat{\models} \langle A^{\mathrm{h}}\theta, \varphi' \rangle}$$

Figure 5.2: Definition of Datalog and application of our framework defining $\hat{\models}$ and $\tau_{\models}$ (the base case concerning facts is included as the case where $m = 0$).

Furthermore, probabilistic entailment needs to be well-formed. We achieve this by defining well-formed entailment $\hat{\models}^*$ using the transformation rule of Equation 4.7, i.e.,

$$\forall A \in \mathit{Atom} : \Phi_A \neq \varnothing \Rightarrow \mathrm{KB} \hat{\models}^* \langle A, \bigvee_{\varphi \in \Phi_A} \varphi \rangle \tag{5.1}$$

where $\Phi_A = \{\varphi \mid \mathrm{KB} \hat{\models} \langle A, \varphi \rangle\}$.

Figure 5.3 contains an elaboration of our example in probabilistic Datalog. It expresses uncertainty about (a) whether "Paris Hilton" is a person or a hotel, (b) whether "Paris" is a place and "Hilton" is a brand but only if they are part of a phrase that is interpreted as a hotel, (c) whether a phrase "Paris" refers to entry g11 or g12 in the gazetteer, and (d) whether or not our rule for determining the country is correct in general.

Observe that both $\langle \mathsf{location}(\mathsf{paris}, \mathsf{pos1}, \mathsf{france}), r{=}1 \wedge y{=}1 \wedge x{=}2 \wedge a{=}1 \rangle$ and $\langle \mathsf{location}(\mathsf{paris}, \mathsf{pos1}, \mathsf{canada}), r{=}1 \wedge y{=}1 \wedge x{=}2 \wedge a{=}2 \rangle$ are entailed for this example. The interpretation of this result is that the "Paris" mentioned in position 1 is located in either France or Canada, but not both (they are mutually exclusive due to a). And that the entailment of a location is uncertain in itself due to $r{=}1 \wedge y{=}1 \wedge x{=}2 \wedge z{=}1$, i.e., only if "Paris Hilton" mentioned

$$\mathsf{type}(\mathsf{paris\_hilton}, \mathsf{pos1\text{-}2}, \mathsf{person}) \overset{x=1}{\leftarrow}$$
$$\mathsf{type}(\mathsf{paris\_hilton}, \mathsf{pos1\text{-}2}, \mathsf{hotel}) \overset{x=2}{\leftarrow}$$
$$\mathsf{type}(\mathsf{paris}, \mathsf{pos1}, \mathsf{place}) \overset{y=1}{\leftarrow} \mathsf{type}(\_, \mathsf{Pos}, \mathsf{hotel}), \mathsf{contains}(\mathsf{pos1}, \mathsf{Pos})$$
$$\mathsf{type}(\mathsf{hilton}, \mathsf{pos2}, \mathsf{brand}) \overset{z=1}{\leftarrow} \mathsf{type}(\_, \mathsf{Pos}, \mathsf{hotel}), \mathsf{contains}(\mathsf{pos2}, \mathsf{Pos})$$
$$\mathsf{gazetteer}(\mathsf{g11}, \mathsf{paris}, \mathsf{france}) \overset{\top}{\leftarrow}$$
$$\mathsf{gazetteer}(\mathsf{g12}, \mathsf{paris}, \mathsf{canada}) \overset{\top}{\leftarrow}$$
$$\mathsf{refersto}(\mathsf{paris}, \mathsf{pos1}, \mathsf{g11}) \overset{a=1}{\leftarrow}$$
$$\mathsf{refersto}(\mathsf{paris}, \mathsf{pos1}, \mathsf{g12}) \overset{a=2}{\leftarrow}$$

$$\mathsf{location}(\mathsf{Ph}, \mathsf{Pos}, \mathsf{C}) \overset{r=1}{\leftarrow}$$
$$\quad \mathsf{type}(\mathsf{Ph}, \mathsf{Pos}, \mathsf{place}), \mathsf{refersto}(\mathsf{Ph}, \mathsf{Pos}, \mathsf{G}),$$
$$\quad \mathsf{gazetteer}(\mathsf{G}, \mathsf{Ph}, \mathsf{C})$$

Figure 5.3: Example of a probabilistic Datalog program.

in positions 1 and 2 refers to a hotel (x=2), whether the inference is correct of the first word "Paris" in a hotel name should be the name of a place (y=1), whether the inference is correct of the second word "Hilton" in a hotel name should be the name of a brand (z=1), and whether the inference of a country is correct by means of looking up the name of a place in a gazetteer (r=1).

114 JudgeD semantics The semantics of JudgeD programs are similar to those of ProbLog, in that JudgeD programs specify multiple traditional Datalog programs. In this section we try to give an intuitive understanding of JudgeD semantics, based on the above formalisation.

A JudgeD program J specifies a multitude of traditional Datalog programs, albeit in a more compact representation. Let W_J be the set of all traditional Datalog programs specified by the JudgeD program. Each partitioning ω^n divides W into n covering disjoint partitions. Each program is labelled with a label ω=v, with ω the partitioning, and v the partition into which the program is placed. This way, every program in W has a set of associated labels, with exactly one label from each partitioning. In other words, labels from the same partitioning are mutually exclusive, and exactly one of them is true for any given Datalog program. For example, given partitionings x^2 and y^2 we can construct all Datalog programs in W by enumerating: $\{x{=}1, y{=}1\}$, $\{x{=}1, y{=}2\}$,

$\{x{=}2, y{=}1\}$, and $\{x{=}2, y{=}2\}$.

A JudgeD program consists of a set of clauses. In JudgeD every clause c_i has an attached propositional sentence φ_i called a descriptive sentence. We use the shorthand $\langle c_i, \varphi_i \rangle$ to denote that sentence φ_i is attached to clause c_i. The descriptive sentence uses partitioning labels, of the form $\omega{=}v$, as atoms to A describe the set of traditional Datalog programs for which the Datalog clause holds: the clause is part of every Datalog program for which φ_i evaluates to true given that the labels attached to the Datalog program are the only labels that are true. For example, the clause $\langle A, x{=}2 \rangle$ is fully defined as follows:

$$A^h \overset{x=2}{\leftarrow} A_1, A_2, \ldots, A_i \tag{5.2}$$

This clause has the normal semantics that A^h holds if A_1 through A_i hold, and only in those Datalog programs for which the descriptive sentence $x{=}2$ holds. Using the previous example partitionings, this clause is part of two Datalog programs specified with the following sets labels: $\{x{=}2, y{=}1\}$ and $\{x{=}2, y{=}2\}$.

To illustrate that a JudgeD project specifies a multitude of normal Datalog programs, observe that the above mentioned entailment location(paris, pos1, france) for the program of Figure 5.3, is valid for the normal Datalog program below belonging to those worlds where $r{=}1 \wedge y{=}1 \wedge x{=}2 \wedge a{=}1 \wedge z{=}1$ holds:

```
type(paris_hilton, pos1-2, hotel) ←
type(paris, pos1, place) ← type(_, Pos, hotel), contains(pos1, Pos)
type(hilton, pos2, brand) ← type(_, Pos, hotel), contains(pos2, Pos)
gazetteer(g11, paris, france) ←
gazetteer(g12, paris, canada) ←
refersto(paris, pos1, g11) ←

location(Ph, Pos, C) ←
   type(Ph, Pos, place), refersto(Ph, Pos, G),
   gazetteer(G, Ph, C)
```

115 JudgeD syntax sample The syntax of a JudgeD program closely resembles traditional Datalog, with the addition of the descriptive sentences. Additionally, the probabilities attached to the labels are included in the syntax.

An example of a simple coin-flip would be:

```
heads(c1) [x=1].
tails(c1) [x=2].

@P(x=1) = 0.5.
@P(x=2) = 0.5.
```

The first two lines establish simple facts and attach sentences to make them mutually exclusive. The third and fourth line contain annotations that attach probabilities to the labels to allow the calculation of answer probabilities. When presented with the query `heads(C)?` the answer `heads(c1)` has a probability of 0.5.

A shorthand for uniform probability distributions is the use of the anotation `@uniform(p)`, where `p` is a partitioning. This shorthand is assigns equal probabilities to all currently defined labels in the partitioning. For the sake of brevity, we often omit probability annotations if they are not needed.

116 Dependencies between clauses The dependencies between clauses can be expressed with descriptive sentences logically combining different labels. Mutual dependency can be expressed by using the same sentence for the clauses. For example $\langle a, \varphi\rangle$ and $\langle b, \varphi\rangle$ describe the situation where the clauses a and b always hold in the same Datalog programs. Implication can be expressed by containment. For example $\langle a, \varphi\rangle$ and $\langle b, \varphi \wedge \psi\rangle$ describes the situation that whenever a is in a Datalog program, then b is too. Mutual exclusivity can be expressed through mutually exclusive sentences. For example, $\langle a, \varphi\rangle$ and $\langle b, \psi\rangle$ are mutually exclusive if $\varphi \wedge \psi \equiv \mathit{false}$.

117 Probability calculation The way of calculating probabilities is defined by the framework (see Section 4.1.3). Any of the generic optimisations discussed in Section 4.4.1 can be applied. But note that in any case, given a JudgeD program J and a query q, the calculated probability of the query answer is equal to enumerating all possible Datalog programs $P \in W_J$, and

summing up the probabilities of each program P for which there is a proof for q.

5.1.2 Example: Maritime Evidence Combination

Recall the maritime evidence combination case presented in Section 1.7.2. It describes the need for combining evidence from different reports for assessing risk properties of incoming ships. A motivating example for the development of JudgeD is its use as reasoning system for this combination of uncertain evidence about maritime data [46]. The case has as ultimate goal the automatic determination of the chance that an observed vessel is engaged in smuggling based on a observations about these vessels. A simplified example of such an observation expressed in Datalog would be the following: `seen("ZANDER", "ROTTERDAM")`. The `seen/2` predicate expresses that a vessel, ZANDER, is seen in a port, ROTTERDAM.

Reasoning about the observations is supported by a knowledge base of factual knowledge about vessels and their attributes. A sample fishing vessel called ZANDER and identified with IMO number 7712767 — the International Maritime Organization number is a unique identifier for the vessel — is described in the knowledge base through the `vessel/1`, `vessel_name/2` and `vessel_imo/2` predicates. The ZANDER is described as:

```
vessel(v0).
vessel_name(v0, "ZANDER").
vessel_imo(v0, 7712767).
vessel_type(v0, stern_trawler).
```

Additional attributes in the knowledge base are described as additional predicates matching the `vessel_???/2` pattern.

The goal is to answer the query `smuggling(V)?` with a set of vessels, each associated with the probability that, given the observations, they are engaged in smuggling.

118 Uncertain facts Facts, both observations and vessel information in the knowledge base, can be uncertain. An example of an uncertain observation

is the interpretation of a verbally reported observation: uncertainty about the observed vessel is easily possible due to a low-quality radio communication. The two interpretations of the spoken report are: `seen("ZANDER", "ROTTERDAM")` and `seen("XANDER", "ROTTERDAM")`. These two observations are mutually exclusive with each other.

Another example would be uncertainty about two observations. For example, if a harbour master receives two reports from different sources about a ship sighted of the coast, there can be doubt about whether these are two ships, or if this is one vessel sighted twice. When he receives a radioed message about a sighting of the vessel `XANDER` and at the same time gets a message about the just sighted `ZANDER`, there are three possible ways of reporting the situation for which it is unknown which is the correct one:

```
report(r1, "XANDER").
```

He makes a single report stating that the vessel `XANDER` was sighted, assuming that the second sighting was actually the same ship, but with an unclearly pronounced name.

```
report(r1, "ZANDER").
```

He makes a single report stating that the `ZANDER` was sighted, confident that the other report was simply a report for the same ship.

```
report(r1, "XANDER").
report(r2, "ZANDER").
```

Alternatively, the harbour master can make two reports. If both names were heard correctly, there are two ships of the coast. In this situation, there is uncertainty about what facts are true.

In conclusion, the harbour master has three options: report one vessel named XANDER (`n=1`), report one vessel named ZANDER (`n=2`), or report them both as separate vessels (`s=2`). In JudgeD this can be expressed as follows:

```
report(r1, "XANDER") [ s=1 and n=1 ].
report(r1, "ZANDER") [(s=1 and n=2) or s=2].
report(r2, "XANDER") [s=2].
```

By creating a partitioning s^2 we effectively describe a choice between Datalog programs: one where the two reports refer to the same vessel, and another where the two reports refer to different vessel. The choice of selecting the name XANDER or ZANDER, represented by the partitioning n^2, is dependent upon `s=1`, as expressed by the conjunction. Complex dependencies can be expressed by combining the `and`, `or` and `not` operations.

119 Uncertain rules Probabilities attached to rules can be interpreted as a form of heuristic. By stating that a rule does not always hold, any answers derived through that rule will take the probability that the rule holds into account. For example, if domain expertise holds that any vessel caught smuggling is likely to be engaged in smuggling again, this can be expressed by the rule: `smuggling(V) :- caught_smuggling(V).` By attaching a probability to this rule, it becomes a heuristic for determining if a vessel is engaged in smuggling.

Dependencies between rules are necessary to express such heuristics with disjunctions in them: if there is a 0.45 chance that ship is smuggling if it is "blue or has an unreadable name" — a purely fictitious heuristic — this is expressed in JudgeD through two separate rules:

```
smuggling(V) :- vessel_blue(V) [h=1].
smuggling(V) :- vessel_name_unreadable(V) [h=1].

@P(h=1) = 0.45.
@P(h=2) = 0.55.
```

These two rules need to be in or out together, as they are two parts of a disjunction that only has meaning as a whole. By introducing partitioning h^2 we effectively divide the all possible worlds between two groups: those labelled h=1 where the heuristic is always correct, and those labelled h=2 where the heuristic is ignored. This is an example of a rule with complex dependencies that is expressible in JudgeD but, as far as we know, not in any other probabilistic logic.

5.1.3 Implementation

In the same way as the framework of Chapter 4 can be applied to the formal foundation of a data model and its query language, a good development strategy is to start from an existing implementation and then introduce sentence manipulation in the appropriate places. Therefore, a basic implementation of Datalog with negation was created first, based on SLG resolution for negative Prolog as described in [19]. The focus of the proof-of-concept implementation of JudgeD is not on raw performance, but on ease of prototyping, as such the system is implemented in Python [35] to allow for quick prototyping of new approaches. The implementation also allows the introduction of native predicates, i.e., predicates that are implemented in Python. Native predicates can be used to pull data from external data sources, such as a relational database or a graph database, into the query answering process. The implementation contains two methods of evaluation: a Monte Carlo approximation and an exact solver.

Monte Carlo Approximation

Monte Carlo approximation for a query q boils down to repeated weighted sampling of a traditional Datalog program P_i from all implicitly specified Datalog programs W_J in the JudgeD program J analogous to what is described in Paragraph 117, and evaluating q for each sampled P_i. Sample weights are calculated by simple multiplication of the probabilities attached to the labels associated with P.

Instead of determining the weights of each Datalog program, a lazy-evaluation scheme is used to construct a set of sampled labels only from those partitionings that are encountered during the search for a proof. This scheme allows the evaluation of q over knowledge bases with enormous amounts of uncertainty, as long as that uncertainty is 'local'. That is, if the uncertainty is expressed as large numbers of partitionings, each with a moderate number of labels.

The implementation features a rudimentary stopping criterion by determin-

ing the root mean square error of the samples observed up till now, and if the error moves below a configurable threshold the approximation is terminated.

The Monte Carlo approximation allows the use of the full expressiveness of negative Datalog, with the lazy-evaluation scheme allowing the application to knowledge bases with large amounts of uncertainty. Furthermore, because of the non-intrusive nature of the scheme, it can easily be applied to other types of solvers. A disadvantage of the Monte Carlo solver is that it will not provide the logical sentence that describes for which Datalog programs the proof holds. It will only provide the probability of the answer.

Exact Solver

In contrast with the Monte Carlo solver, the Exact solver determines the exact sentence φ_a describing in which Datalog programs the proof for the answer a was found. This is done based on the knowledge that for any answer the resolution proof can be restricted to a linear sequence of clauses $c_1, c_2, \ldots, c_i$, with attached sentences $\varphi_1, \varphi_2, \ldots, \varphi_i$. The sentence for the answer follows from the needed clauses as:

$$\varphi_a = \bigwedge_{n=1}^{i} \varphi_n \tag{5.3}$$

If the sentence φ_a is consistent, then answer a can be proven in all Datalog programs for which the sentence holds. An inconsistent sentence shows that there are no Datalog programs contained within the JudgeD program for which there is a valid proof.

120 Sentence construction and unification Efficient construction of this sentence is done by constructing partial sentences during SLG resolution, i.e., unification, of two clauses G and C. The partial sentence φ_A for the resolvent A is equal to the conjunction of the sentences associated with G and C: $\varphi_A = \varphi_G \wedge \varphi_C$. If this sentence is inconsistent this means that G and C are not unifiable because there is no Datalog program for which this proof will hold.

If a new fact is discovered during the search it is only necessary to expand on it if it is not subsumed by an already discovered fact. While Datalog has no functions, and thus no functional subsumption, the introduction of descriptive sentences creates a different kind of subsumption. A new fact $\langle f, \varphi \rangle$ is subsumed by an already known fact $\langle a, \psi \rangle$ if $\varphi \wedge \psi \equiv \psi$. If this is the case, the new fact does not add new knowledge to the already expanded knowledge base, because any proof that leads to the new fact comes from already explored Datalog programs.

121 Determining sentence subsumption The efficient detection of sentence subsumption is done through the use of binary decision diagrams [13]. A binary decision diagram is a graphical representation of a boolean function over a number of variables. Given a complete ordering over the variables a Reduced Ordered Binary Decision Diagram, more commonly known simply as a BDD, provides a canonical representation of the boolean function. Therefore, checking the subsumption $\varphi \wedge \psi \equiv \varphi$ is the same as checking if φ and $\varphi \wedge \psi$ are represented by the same BDD.

A BDD can be constructed by starting with a binary decision tree in which all non-leaf nodes represent variables and all leaf nodes represent either 1 or 0. Non-leaf nodes have a 'high' and a 'low' child. Each path from the root to a leaf represents a full assignment of truth values to each variable, with variables encountered in the order determined by the full ordering. A BDD can be constructed from this tree by merging isomorphic subgraphs and reducing redundant nodes until no further reduction is possible. An example of a BDD for the function $\varphi = (x{=}1 \wedge y{=}2) \vee z{=}1$ can be seen in Figure 5.4. This figure also showcases the subsumption check: assuming $\psi = z{=}1$ the figure also shows the BDD for $\varphi \wedge \psi$.

The current exact implementation is restricted to positive Datalog, and does not yet calculate probabilities. See Section 5.1.5 for a discussion on extension to negative Datalog, and on two promising directions for probability calculation.

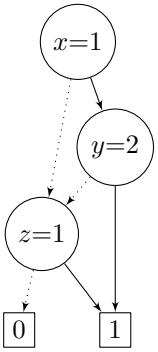


Figure 5.4: Example of a BDD for the sentence $\varphi = (x{=}1 \wedge y{=}2) \vee z{=}1$. Solid edges are high, dotted edges are low.

5.1.4 Related Work

The semantics of JudgeD can be seen as an extension to the semantics of ProbLog. In ProbLog [92] each clause has an attached probability that it is true. These probabilities are assumed to be independent. We extend this semantic by decoupling the probabilities from the clauses through the descriptive sentences, allowing the expression of complex dependencies. Furthermore, where the ability to assign probabilities to rules has to be exercised with caution in ProbLog — because, as De Raedt et al. [92] state "the truth of two clauses, or non-ground facts, need not be independent, e.g. when one clause subsumes the other one" — this is not a concern in JudgeD where these clauses can be given multiple labels.

The probabilistic Datalog pD [37] also assumes independent probabilities, and allows the definition of sets of disjoint events. In this way it is possible to model arbitrary dependencies. This can be done by providing the disjoint probabilities for all possible combinations of dependent events. In practice, the required enumeration of all possible combinations makes it an infeasible solution to expressing complex dependencies.

MCDB [58] uses a Monte Carlo approach to allow query answering over a sampled database, where they apply their concept of tuple-bundles to speed

up the process. JudgeD uses a conceptually similar method through the lazy-evaluation scheme, which answers the query by monotonically constricting the answer to the set of Datalog programs in which a proof can be found.

5.1.5 Future Research

There are several areas for improvement and investigation concerning JudgeD. Of specific interest is the computation and approximation of probabilities.

122 ProbLog-based implementation JudgeD is a probabilistic Datalog derived from the framework described in Chapter 4. The decoupling between descriptive sentence and Datalog clause closely adheres to this framework. However, with ProbLog's proven performance on top of YAP-Prolog,[64] an implementation based on this system is a promising and open topic of investigation.

123 Exact probability calculations Exact probability calculation for sentences in Disjunctive Normal Form (DNF) is described by [66]. They propose an algorithm and heuristic to break down the DNF sentence into independent subsentences, which allows computation of the exact probability. Another venue of investigation of probability calculation is ProbLog's approximation [92]. ProbLog demonstrates both a BDD-based probability calculation for the exact probability and an approximation algorithm that can be applied during the computation of the SLD tree. Since the currently used solver is based on SLG [19], which is a successor of SLDNF, this direction seems to be valuable.

The current implementation of the exact solver does not support negative Datalog. Our work in the maritime evidence combination case has shown the need for negation in real-life applications. Further investigation is needed to apply the formalism described in Chapter 4 to negative Datalog to allow a principled implementation of the exact solver for SLG resolution to support negation.

124 Generation of partitionings for infered worlds A different direction is the extension of JudgeD to allow for generalized probabilities. Currently, the modelling of two coin flips requires the explicit declaration of a second partitioning. For example, a single coin flip can be modelled by: $\{ \langle coin(c1), true\rangle$ $\langle heads(C) \leftarrow coin(C), x{=}1\rangle$, $\langle tails(C) \leftarrow coin(C), x{=}2\rangle$ $\}$ with a single partitioning x^2 describes how the coin *c1* can go either heads or tails. Two coin flips must be made explicit with the addition of a new partitioning y^2. The simple addition of $coin(c2)$ to the previous scenario will result in x^2 representing a 'universal' coin flip: either all coins land on heads, or all coins land on tails. Extending the modelling of probabilities to allow the specification of implicit partitionings, i.e., the specification of "one partitioning per X for all answers of $coin(X)$", together with their probability mass functions may improve the way JudgeD can be applied to certain real-world problems such as entity resolution.

125 Usage of external probabilistic data sources JudgeD has the option to use knowledge from external data sources, such as relational databases or graph databases, through native predicates. At the moment of writing, native predicates must be deterministic to leverage the full expressiveness of JudgeD, native predicates have to be extended to allow them to interface with probabilistic relational databases and other probabilistic data sources.

5.1.6 Conclusions

JudgeD is a proof-of-concept probabilistic Datalog based on the formalism from Chapter 4. As such it is the first part in the validation of the data model independence properties of the framework. The direct application of the framework resulted in a probabilistic Datalog that is more expressive than other probabilistic logics in that it can express complex dependencies between arbitrary clauses (i.e., both facts and rules).

The JudgeD implementation can connect to external data sources through native predicates, and supports negative Datalog based on SLG resolution. We have presented a Monte Carlo approximation for calculating answer probabilit-

ies, and presented an exact solver that works for positive Datalog. The key contribution of JudgeD is the ability to express dependencies between arbitrary clauses, including both facts and rules in such dependencies.

There are several venues for future investigation, including improved probability calculation algorithms inspired by MayBMS and ProbLog, the use of external probabilistic data sources, and the addition of generalized probabilities to express independent probabilities associated with repeated or plural events.

JudgeD is released under the MIT license. The complete system can be obtained from: `https://github.com/utdb/judged`.

5.2 Probabilistic XML / XPath

The XML data model effectively describes a heterogeneous tree. Nodes in the tree can be of many node kinds such as elements, text nodes, attributes, etc. XPath is a query language over the XML data model.

The XPath query language allows one to describe a navigational structure over a tree, with the results of the query being all elements that can be reached by following the navigational structure. As an XML query language used in both industry and academia, XPath has evolved from version 1.0 through version 3.0 [27, 10, 103]. In this time it has seen significant change both in its expressiveness and underlying philosophy.

In this section we present a probabilistic XPath created by applying our framework to the XML data model with XPath as query language.

126 A representative subset of XPath For the purposes of our investigation of probabilistic XPath, we define a representative subset of XPath to which we apply our probabilistic framework.

We view an XPath query as a sequence of navigation steps consisting of an axis and a node test. Intuitively the axis describes the direction of the step while the node test describes a filter to select nodes that are acceptable. Optionally, a step may have one or more predicates that further refine the acceptable nodes.

To ease the writing of XPath queries a number of shorthands are defined. For example, the query `//a/b[@p]` is the shorthand representation of:

```
/descendant-or-self::a/child::b[attribute::p]
```

This XPath query describes a two-step sequence. The first step is over the descendant-or-self axis and selects nodes with name "a", and the second step selects direct children named "b" that have an attribute called "p".

During query answering, the navigational steps in the XPath are applied in sequence. Each step results in a set of nodes that forms the input for the next step. The result of the step is determined by finding all nodes from the axis for which both the node test and all predicates hold.

In contrast with full XPath where predicates can be arbitrary expressions, our subset of XPath only includes path based predicates.

5.2.1 Framework applied to XPath

Before we go into the application of our framework to XPath, we make a short detour towards the probabilistic XML data model itself.

Application of the framework described in Chapter 4 on the XML data model yields a probabilistic XML. In this probabilistic XML the nodes in the document are the assertions in our framework. In this way, a probabilistic XML document describes multiple certain XML documents.

The descriptive tuple $\hat{n} = \langle n, \varphi \rangle$ describes an XML node and the associated descriptive sentence. Since each possible world describes one, and only one, tree, a node can only exist if its parent exists, hence in effect all of its ancestors must exist. We therefore distinguish between a node's descriptive sentence φ and its *total descriptive sentence* $\varphi^{\uparrow}$:

$$\varphi^{\uparrow}(n) = \varphi(n) \bigwedge_{x \in \text{ancestors}(n)} \varphi(x) \tag{5.4}$$

where ancestors(...) is a function that produces the ancestors of an element, i.e., its parent, its parent's parent, etc. all the way up to the root.

This section continues with a definition of XPath, and an application of our framework to produce a probabilistic XPath. We present an implementation based on a translation to XQuery in Section 5.2.3, followed immediately with a discussion of optimisation in Section 5.2.4. Finally, we discuss related work in Section 5.2.5.

127 Definition of XPath We postulate disjoint sets *Node*, *Axis*, *Test* and *Pred* as the sets of *XML nodes*, *axis functions*, *node tests* and *predicates*, respectively. Let a *node* $n \in Nodes$ be either an element, text node, attribute, processing instruction or comment as allowed by the XML data model. An *axis functions* $a \in Axis$ is a function $a : Node \to \mathbb{P}Node$ mapping a node to a set of nodes according to one of the defined XPath navigational axes. Let $p \in Pred$ be a path-based predicate function $p(n) = (\text{evaluate}(X_p, \{n\}) \neq \varnothing)$ where X_p is the relative XPath expression of the predicate, n is the context node, and evaluate is the XPath evaluation function.

A *step* $s \in Step$ defined as $s = (a, t, p_1, \ldots, p_j)$ consists of an axis function a together with a node test t and zero or more predicates p_i. An XPath *expression* $X = (s_1, \ldots, s_n)$ is a sequence of steps. The function

$$\text{step}(s) : Step \to \mathbb{P}Nodes \to \mathbb{P}Nodes \tag{5.5}$$

applies a single step from an XPath expression and is defined as:

$$\frac{\begin{array}{c} s = (a, t, p_1, \ldots, p_j) \\ n \in a(c) \quad c \in C \\ t(n) \wedge p_1(n) \wedge \ldots \wedge p_j(n) \end{array}}{n \in \text{step}(s)(C)} \tag{5.6}$$

The evaluation of a complete XPath expression is a sequential application of

steps:

$$\text{evaluate}(X, C) = \text{evaluate}(s_1, \ldots, s_n, C) \tag{5.7}$$
$$= (\text{step}(s_n) \circ \text{step}(s_{n-1}) \circ \ldots \circ \text{step}(s_1))(C) \tag{5.8}$$

where X is a sequence of steps forming an XPath and C is the set of context nodes, i.e., a singleton set containing either the root node for absolute paths, or the context node for relative paths.

128 Probabilistic XPath The semantics of probabilistic XPath follow directly from applying the framework to XPath. The descriptive sentences attached to the XPath answers produced by $\widehat{\text{evaluate}}(X, C)$ (the probabilistic counterpart of $\text{evaluate}(X, C)$) are the conjunction of total sentences of all the nodes visited while navigating the tree structure. Our framework proposes the definition of a τ transformation function to explicitly define sentence construction for answer sentences. For the $\text{step}(s)$ function τ_{step} is defined as:

$$\varphi_1, \ldots, \varphi_m \overset{\tau_{\text{step}}}{\mapsto} \bigwedge_{i \in 1..m} \varphi_i \tag{5.9}$$

where φ_i is a descriptive sentence.

The probabilistic version of $\text{evaluate}(X, C)$, denoted as $\widehat{\text{evaluate}}(X, C)$, is defined in terms of the probabilistic $\widehat{\text{step}}$ function:

$$\widehat{\text{evaluate}}(X, C) = \widehat{\text{evaluate}}(s_1, \ldots, s_n, C) \tag{5.10}$$
$$= (\widehat{\text{step}}(s_n) \circ \widehat{\text{step}}(s_{n-1}) \circ \ldots \circ \widehat{\text{step}}(s_1))(C) \tag{5.11}$$

where, again, X is a sequence of steps forming an XPath and C is the set of context nodes.

In $\widehat{\text{step}}$ each predicate p_i is evaluated through the $\widehat{\text{evaluate}}$ function: the predicate test $p_i(\hat{n})$ visits nodes while navigating the XML tree to find a matching path. Each match for the predicate's path X_{p_i} is a navigation

$$\frac{\begin{array}{c} s = (a, t, p_1, \ldots, p_j) \\ n \in a(c) \quad c \in C \\ t(n) \wedge p_1(n) \wedge \ldots \wedge p_j(n) \end{array}}{n \in \mathrm{step}(s)(C)} \xrightarrow{\mathrm{step}} \frac{\begin{array}{c} \varphi_1, \ldots, \varphi_m \overset{\tau_{\mathrm{step}}}{\mapsto} \bigwedge_{i \in 1..m} \varphi_i \\ s = (a, t, p_1, \ldots, p_j) \\ \hat{n} = \langle n, \varphi_n \rangle \quad \hat{n} \in a(c) \quad \langle c, \varphi_c \rangle \in C \\ t(n) \wedge p_1(\hat{n}) \wedge \ldots \wedge p_j(\hat{n}) \\ \varphi'_n = \varphi^{\uparrow}(n) \quad \varphi'_c = \varphi^{\uparrow}(c) \\ \forall_{i \in 1..j} : \hat{E}_{p_i} = \widehat{\mathrm{evaluate}}(X_{p_i}, \{\hat{n}\}) \\ \forall_{i \in 1..j} : \varphi'_{p_i} = \bigvee_{\hat{e} \in \hat{E}_{p_i}} \varphi(\hat{e}) \\ \varphi' = \tau_{\mathrm{step}}(\varphi'_c, \varphi'_e, \varphi'_{p_1}, \ldots, \varphi'_{p_j}) \end{array}}{\langle n, \varphi' \rangle \in \widehat{\mathrm{step}}(s)(C)}$$

Figure 5.5: The full definition of $\widehat{\mathrm{step}}$.

through the XML tree that validates the predicate, the set of matches being:

$$\hat{E}_{p_i} = \widehat{\mathrm{evaluate}}(X_{p_i}, \{\hat{n}\}) \tag{5.12}$$

where $\hat{n}$ is the context node, and X_{p_i} the predicate path for the predicate p_i. To take into account all possible worlds in which the predicate holds, the sentence φ_{p_i} attached to the predicate evaluation is a disjunction of the sentences of all navigations for which the predicate is true. The sentence associated with the predicate is constructed by:

$$\varphi_{p_i} = \bigvee_{\hat{e} \in \hat{E}_{p_i}} \varphi(\hat{e}) \tag{5.13}$$

The definition of $\widehat{\mathrm{step}}$ is analogous to the step function, with the addition of sentence construction through τ_{step} and combined with the sentence construction for predicate matches. The full definition can be seen in Figure 5.5.

5.2.2 Maritime Evidence Combination

To show the evaluation of an XPath expression on an actual XML document, we revisit the maritime evidence combination example (Section 1.7.2) once more. In Section 5.1.2 we described the scenario where the harbour master receives two reports.

In this scenario, the harbour master receives two messages from different sources about a ship sighted of the coast. Having received the radioed message about a sighting of the `XANDER`, and another one at the same time about the `ZANDER`, he is uncertain about the actual situation and is left with three different ways to report the situation: there is only one vessel called `XANDER`, there is only one vessel called `ZANDER`, or there are two vessels.

129 A report in probabilistic XML Which one of these outcomes is reported depends upon the choices made by the harbourmaster. The harbour master's first choice (represented by partitioning s^2) is whether there actually are two vessels, or if the two messages he received refer to the same vessel. If he concludes that there is only one vessel, the second choice (represented by partitioning n^2) is the selection of the correct name.

The probabilistic XML variant of the report by a harbour master in the Port of Rotterdam, is expressed as follows:

```
<report id="1">
    <location id="2">
        <port id="3">ROTTERDAM</port>
    </location>
    <vessels id="4">
        <vessel id="5" p="s=1 and n=1">XANDER</vessel>
        <vessel id="6" p="(s=1 and n=2) or s=2">ZANDER</vessel>
        <vessel id="7" p="s=2">XANDER</vessel>
    </vessels>
</report>
```

Descriptive sentences are attached to the XML elements through the `p` attribute. All nodes without this attribute are assumed to have the attached sentence

$\varphi = \top$ and exist in every certain tree.

130 Expressing a probabilistic XPath query The coast guard collects the reports from the different harbour masters, and combines them into a single collection. Then, another division of the coast guard can use this collection to make informed decisions on the deployment of patrols and inspections. To make this decision, they want the answer to questions "Which ports reported seeing the vessel XANDER of the coast?". This question can be expressed in XPath as follows:

```
//report[vessels/vessel="XANDER"]//port
```

Note that this query uses a predicate that is not solely path-based, while in our formalisation of XPath we have only discussed path-based predicates. This text-equivalency predicate is in effect a path-based predicate to test for the existence of a text node, followed by a test of the node's text contents. The implementation discussed in Section 5.2.3 understands this kind of predicate.

Recall that, in Paragraph 127, we formalise an XPath expression as sequence of steps. Each step consists of a navigation axis a, a node test t and optionally some predicates p:

$$s = (a, t, p_1, \ldots, p_j) \tag{5.14}$$

Before we can illustrate the evaluation of the above query, we must first express it in terms of steps.

The first step s_1 selects all reports from the collection. Formally, the shorthand `//report` expands to two separate steps. We can collapse these steps, without loss of generality, into a single step using the *descendant* axis and testing each node on being a `report`. The predicate p_1 that only those reports that mention the vessel "XANDER" is made explicit later on. The first step of the query is:

$$s_1 = (\mathit{descendant}, \lambda n : \text{“}n \text{ is a } \texttt{report}\text{”}, p_1) \tag{5.15}$$

The second step s_2 selects all ports, that is, it uses the *descendant* axis and tests each node on being a `port`:

$$s_2 = (\mathit{descendant}, \lambda n : \text{“}n \text{ is a } \texttt{port}\text{”}) \tag{5.16}$$

These two steps together form the XPath expression $X = (s_1, s_2)$. The mentioned predicate p_1 is defined as a two-step sequence:

$$x_1 = (\mathit{child}, \lambda n : \text{“}n \text{ is a } \texttt{vessels}\text{”}) \tag{5.17}$$

$$x_2 = (\mathit{child}, \lambda n : \text{“}n \text{ is a } \texttt{vessel}\text{”}, \text{“Text content is XANDER”}) \tag{5.18}$$

With the complete query defined it can be used together with a context C as the input to the $\widehat{\text{evaluate}}$ function.

131 Evaluating a probabilistic XPath expression The $\widehat{\text{evaluate}}$ function works by applying each step in the expression to C in sequence. This means that, for the expression we have defined above, the $\widehat{\text{evaluate}}$ function boils down to:

$$\widehat{\text{evaluate}}(X, C) = (\widehat{\text{step}}(s_2) \circ \widehat{\text{step}}(s_1))(C) \tag{5.19}$$

We illustrate the evaluation of this function by applying it to a collection containing the report from Paragraph 129, which we assume is the only report about the XANDER. We refer to each node in the document by it's identifier as stated by the `id` attribute.

We start out with a context of $C = \{\langle r, \top \rangle\}$. The root of the collection identified by r, with a descriptive sentence of $\top$ to state that the root always exists.

The first step $\widehat{\text{step}}(s_1)$ produces $C' = \{\langle 1, (s{=}1 \wedge n{=}1) \vee s{=}2 \rangle\}$. It does so by applying the axis function *descendant* which produces all the nodes below the root, and then filtering with $\lambda n :$ “n is a `report`”, which narrows it down to only the reports.

To further narrow down the selection, the first predicate is tested on each report. This predicate tests for the existence of a `vessel` node that has "XANDER" as its text content. Each possible way of fulfilling the predicate is relevant, hence the term:

$$\hat{E}_{p_1} = \widehat{\text{evaluate}}(X_{p_1}, \{\langle 1, \top\rangle\}) \tag{5.20}$$

$$= \{\langle 5, s{=}1 \wedge n{=}1\rangle, \langle 7, s{=}2\rangle\} \tag{5.21}$$

Which is then combined into φ'_{p_1} through:

$$\varphi'_{p_1} = \bigvee_{\hat{e} \in \hat{E}_{p_1}} \varphi(\hat{e}) \tag{5.22}$$

$$= (s{=}1 \wedge n{=}1) \vee s{=}2 \tag{5.23}$$

The τ_{step} function then combines this sentence with the total sentence for the report itself and the total sentence for the context node, resulting in the sentence shown in C'.

The second step $\widehat{\text{step}}(s_2)$ produces $C'' = \{\langle 3, (s{=}1 \wedge n{=}1) \vee s{=}2\rangle\}$. The axis function for *descendant* and the node test λn : "n is a `port`" narrow down the selected nodes to a single option in the report. The τ transformation function then combines the sentences from the context produced by the first step, and the sentence for the `port` node itself to produce the final sentence. So, the result of the query "Which ports reported seeing the vessel XANDER of the coast?" is the node `<port id="3">Rotterdam</port>` but only if $(s{=}1 \wedge n{=}1) \vee s{=}2$, i.e., if the original sightings referred to one ship ($s{=}1$) and the right name was "XANDER" ($n{=}1$), or the original sightings were two ships ($s{=}2$) which included a ship called "XANDER".

5.2.3 Implementation

The evaluation of a probabilistic XPath on an XML document with embedded descriptive sentences means constructing descriptive sentences for each possible answer.

The proof-of-concept implementation expects the XML document to host the descriptive sentences in `p` attributes. Nodes without such an attribute are assumed to have the attached sentence $\varphi = \top$, and therefore exist in all possible worlds.

132 Implementation by translation We use the language XQuery to create a proof-of-concept implementation of the $\widehat{\text{evaluate}}$ function. XQuery [34] is an expression language featuring specialised operations to extract information from an XML document in a functional side-effect free manner.

The evaluation of probabilistic XPath can be implemented with a normal XQuery program that takes into account the descriptive sentences in the `p` attributes. The produced XQuery programs can then be evaluated on any XQuery engine such as BaseX [42].

The translation of an XPath expression to the possible-world aware XQuery expression is performed by a Python implementation of an XPath parser and translator. The steps are recursively translated into a nested XQuery expression. For example, the expression `/a/b` is translated to:

```
for $step1 in fn:root()/child::a
return
    for $step2 in $step1/child::b
    return
        <match p="{ pxp:conjunction( (pxp:sentence($step2)) ) }">
            { $step2 }
        </match>
```

The first context node for the evaluation is produced by the standard function `fn:root()`, which produces the root node of the document. For brevity, the preamble is omitted. The preamble provides the trivial definitions of the sentence construction functions `pxp:conjunction` and `pxp:disjunction`, and the implementation of $\varphi^{\uparrow}(n)$ as `pxp:sentence`.

Predicates are translated by including them in the 'For, Let, Where, Order by, Return' expression (commonly referred to as the FLWOR expression) to ensure only matching nodes are inspected, and then building the descriptive

sentence for the predicate by explicitly translating the predicate's XPath expression. For example, the expression `/a[c]` translates to:

```
for $step1 in fn:root()/child::a[child::c]
let $pred1 := pxp:disjunction(
    for $step2 in $step1/child::c
    return
        pxp:conjunction( (pxp:sentence($step2)) )
)
return
    <match p="{ pxp:conjunction( (pxp:sentence($step1), $pred1) ) }">
        { $step1 }
    </match>
```

The `$pred1` variable is bound to the disjunction of all the sentences of each possible alternative match for the predicate, and the whole disjunction is used in the final sentence.

5.2.4 XML- and XPath-specific optimisation

The proof-of-concept implementation described so far is a straight-forward implementation of the $\widehat{\text{evaluate}}$ function. Next to enabling the creation of probabilistic data models, the framework also inspires several optimisations in the implementation. Some of these optimisations are general to all data models and query languages, and some are specific to a data model. We present one such XML- and XPath-specific optimisation here.

133 Optimisation based on data model properties The hierarchical nature of XML makes it possible to optimise the construction of descriptive sentences during the evaluation of an XPath query. Because of the implied hierarchy, which is also expressed by $\varphi^{\uparrow}$, parts of the sentences are implied by other parts, so they need not even produced in the construction of disjunctions and conjunctions.

Lemma 5.1 (Inclusion of ancestor sentences) *The hierarchical nature of the XML data model ensures that the total descriptive sentence of a node always implies the total sentence of any ancestor:*

$$\forall n_1, n_2 : n_2 \in \text{descendants}(n_1) \implies (\varphi^{\uparrow}(n_2) \implies \varphi^{\uparrow}(n_1)) \tag{5.24}$$

Where n_1 and n_2 are nodes from the same probabilistic XML document.

The notion formalised in Equation 5.1 can be used to optimise the generated XQuery statement. For example, in the expression `//a/b` we can safely ignore the sentence for matching `a` elements, since they are always included in their `b` children's sentences.

Determining which steps' sentences can be ignored requires investigation of the shape of the navigational structure described by an XPath. This investigation is done based on the XPath's steps. We define four directions: *up* ($\uparrow$), *down* ($\downarrow$), *sideways* ($\leftrightarrow$), *stay* ($\bullet$); every XPath navigation axis can be mapped to one of these directions as follows:

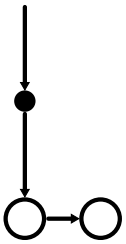
- *up* ($\uparrow$): ancestor, ancestor-or-self, parent,
- *down* ($\downarrow$): attribute, child, descendant, descendant-or-self,
- *sideways* ($\leftrightarrow$): following, preceding, following-sibling, preceding-sibling,
- *stay* ($\bullet$): self.

Additionally, we define the pseudo-directions *start* ($\diamond$) and *end* ($\times$) to facilitate the investigation of the first and last steps of an XPath. The sequence of directions of an XPath can be derived by:

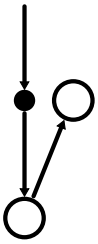
$$\text{directions}(s_1, \ldots, s_n) = (\diamond, \text{d}(s_1), \ldots, \text{d}(s_n), \times) \tag{5.25}$$

Where $\text{d}(s)$ is a function that produces the direction for the given step based on the axis of the step. Some examples of XPath expression and their navigational structures and sequence of directions can be seen in Figure 5.6.

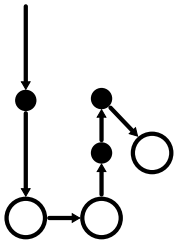
(a) Simple downward structure produced by `a/descendant-or-self::b`. Sequence of directions: $\diamond \downarrow\downarrow \times$



(b) Sideways movement of the expression `a/b/following-sibling::c`. Sequence of directions: $\diamond \downarrow\downarrow\leftrightarrow \times$



(c) Upwards movement produced by `a/b[ancestor::c]`. Sequence of directions: $\diamond \downarrow\downarrow\uparrow \times$



(d) Complex navigational structure showcasing a down-up-down motion. Sequence of directions: $\diamond \downarrow\downarrow\leftrightarrow\uparrow\uparrow\downarrow \times$

Figure 5.6: Four examples of navigational structures and their sequence of directions. Closed nodes represent irrelevant results, and large open nodes represent relevant results.

134 Relevant and irrelevant results An optimisation based on the hierarchical nature of XML is to reduce the number of sentences that are considered relevant: any navigational step for which the result's sentences are implied by the results of a following step can be considered an irrelevant result. The sentences of irrelevant results can safely be ignored. To determine if a result is an irrelevant result or relevant result the direction of the next step is used.

The *stay* (•) direction requires special handling in that any occurrences of this direction are skipped when inspecting the direction of the next step. The complete matrix of possible direction changes is shown in Table 5.1. This optimisation through ignoring redundant sentences applies equally well to path based predicates and to the main path. The introduction of the *start* (⋄) direction allows the investigation and optimisation of relative paths.

For example, observe the XPath expression `a/b/following-sibling::c`. The navigational structure of this expression can be seen in Figure 5.6b. Expanding the short-hand notation makes all the axes explicit and turns the previous expression into the following equivalent expression:

```
child::a/child:b/following-sibling::c
```

Both `child` axes are *down* (↓), and the `following-sibling` axis is *sideways* (↔). Therefore the sequence of directions for this expression is ⋄ ↓↓↔ ×. This sequence of directions can be broken down into four pairs of a current direction

	↑	↓	↔	×
⋄	⋄ ↑	⋄ ↓	⋄ ↔	⋄ ×
↑	↑↑	↑↓	↑↔	↑ ×
↓	↓↑	↓↓	↓↔	↓ ×
↔	↔↑	↔↓	↔↔	↔ ×

Table 5.1: Direction matrix listing combinations of navigational directions: current step's direction (vertical) versus the next step's direction (horizontal). Combinations highlighted in green indicate relevant results. The symbols represent directions *up* (↑), *down* (↓), *sideways* (↔), *start* (⋄), and *end* (×).

and a next direction. Each of these pairs is related to a single step in the expression, with the *start* ($\diamond$) direction related to the context node because this is a relative path. These pairs can be looked up in Table 5.1 to determine the relevance of the step's sentence:

1. The context node with ($\diamond \downarrow$): irrelevant result
2. The step `child::a` with ($\downarrow\downarrow$): irrelevant result
3. The step `child::b` with ($\downarrow\leftrightarrow$): relevant result
4. The step `following-sibling` with ($\leftrightarrow \times$): relevant result

This means that only the sentences for the last two steps need to be taken into account.

135 Sentence Optimisations In addition to these data-model based optimisations there is also the option of optimisation through sentence manipulation. For example, common conjunctive subsentences in a larger conjunction can be factored out during the construction of the sentence. These manipulations are in part motivated by the properties of the data model, but are rooted in well-explored propositional logic techniques.

5.2.5 Related Work

The work related to the probabilistic XML and XPath we created with the application of our probabilistic framework fits into two categories: work on probabilistic XML, and work on querying of probabilistic XML.

Among the works that inspired the investigation of XPath as a validation case for querying over probabilistic XML, three deserve special note. The work by van Keulen et al. on data integration with probabilistic XML [63] for providing a use case, and the work by Hollander et al. on storing and querying probabilistic XML using a probabilistic relational DBMS [55]. The master's thesis by Stapersma [107] further explored the relation between probabilistic XML and probabilistic relational DBMS.

For an excellent overview of previous work on probabilistic XML, and a comprehensive analysis of the expressive power of the different families of probabilistic XML, we refer to [2] by Abiteboul et al. They distinguish between families of probabilistic XML through the presence of five types of distributional nodes. A distributional node indicates that there is uncertainty about the presence of its children, and in this way introduces uncertainty into the document.

Abiteboul et al. refer to these families of probabilistic XML documents as $\mathbf{PrXML}^C$ where C is a subset indicating the presence of certain kinds of distributional nodes, i.e., $C \subseteq \{\mathsf{det}, \mathsf{mux}, \mathsf{ind}, \mathsf{exp}, \mathsf{cie}\}$.

To create a certain instance of an **PrXML** document, i.e., to construct a single possible world, all distributional nodes replaced with zero ore more of their children. Which children are selected depends on the node type. The distributional node types each offer a different way of expressing uncertainty:

det Deterministic nodes select all of their children with certainty. Each child node of a deterministic distributional node has an implicit probability of 1.

mux The mutual exclusion distributional node selects at most 1 of its children. The sum of probabilities assigned to the children of a mutual exclusion node may be less than 1, leaving the option that no replacement is selected.

ind The independent distributional node assigns a probability to each child, and determines the presence of each child in the instance independently from the others.

exp The explicit distributional node defines a probability per subset of child nodes, and selects a single subset to be included in the instance. The empty set $\varnothing$ is an allowed subset.

cie The *cie* distributional node selects children based on the truth value of a conjunction of independent events. These events are global for the

instanced document, meaning that this is the only distributional node that can correlate their choices by sharing events.

Abiteboul et al. show that $\mathbf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$ is the most expressive family of probabilistic XML. If the probabilistic XML produced by the application of our framework is restricted to allow only conjunctive descriptive sentences of positive terms this variant can be mapped to the $\mathbf{PrXML}^{\{\mathsf{cie}\}}$ family. Based on this, it follows that our probabilistic XML is at least as expressive as other models in this family. Further investigation of the expressive power of our variant of probabilistic XML is needed to determine how it relates to the $\mathbf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$ family.

5.3 Probabilistic SQL: MayBMS

As the third case in the validation of data model independence of our framework, we apply it to SQL. SQL is based on the formalisation of relational algebra. Relational Algebra is the underpinning of relational databases. It allows the expression of operations on data structured as relations containing tuples. The tuples in a relation are uniform and comply to the relation's schema which is defined as a set of attributes.

136 Placement of uncertainty Several relational probabilistic databases have been developed. Relational probabilistic database systems that, to a certain degree, have outgrown the laboratory bench include: MayBMS [56, 5], Trio [83], and MCDB [58] as a prominent example of a Monte Carlo approach.

MayBMS and Trio focus on tuple-level uncertainty, that is, probabilities are attached to tuples, and mutually exclusive sets of tuples are defined. MCDB focuses on attribute-level uncertainty where a probabilistic distribution captures the possible values for the attribute. Analogous to the probabilistic logics, certain constraints are imposed.

In Trio probabilities are attached to tuples in exclusive sets (sets of mutually exclusive tuples) of which at most one is selected. MCDB supports expressing

correlation between attributes through correlated sample functions. MayBMS allows the expression of mutual exclusivity and mutual dependency.

In the application of our framework to the relational model, we focus on tuple-level uncertainty.

137 Outlook In the next section we discuss the application of our framework to the relational data model. Subsequently, Section 5.3.2 gives an illustrative example where the ambiguous result of named entity extraction and disambiguation is stored in probabilistic relations. Finally, in Section 5.3.3 we show that MayBMS is an implementation of the produced probabilistic relational model albeit under certain restrictions. We follow up with a validation of the applicability of the probabilistic relational model in real-world situations in Chapter 6.

5.3.1 Framework applied to relational algebra

We start out the application of the framework with a formal definition of relational algebra. Next we show how to apply our framework to relational algebra by presenting probabilistic versions of the core operations of relational algebra.

138 Definition of Relational Algebra We postulate a set of *attribute domains Int*, *Bool*, *String*, etc. Let $R(at_1, \ldots, at_n) \subseteq \mathrm{dom}(at_1) \times \cdots \times \mathrm{dom}(at_n)$ be a *relation* containing *relational tuples* $r \in R$ with attributes $at_1, \ldots, at_n$ where $\mathrm{dom}(at_i)$ denotes the domain of at_i ($i \in 1..n$).

Operations include the usual set operations *union* ($\cup$), *intersection* ($\cap$), and *difference* ($\setminus$) together with *selection* (σ), *projection* (π), *cartesian product* ($\times$), and *join* ($\bowtie$).

The usual restrictions apply, for example, set operations require the operands to have the same attributes. We define the relational operators alongside the probabilistic ones below for easy comparison.

139 Probabilistic Relational Algebra Using our framework, we obtain probabilistic relational algebra by viewing relational tuples as assertions. Hence, instead of normal tuples, relations now contain descriptive tuples.

For each operator $\oplus$, we define $\hat{\oplus}$ in terms of $\oplus$ and $\tau_{\oplus}$ where the latter maps descriptive sentences of the operands to a descriptive sentence of the result. We then 'weave' the application of $\tau_{\oplus}$ into the definition of the original non-probabilistic operators $\oplus$. Let $A(R) = \{a(\hat{a}) \mid \hat{a} \in R\}$ be the set of assertions (i.e., relational tuples) from a probabilistic relation R.

Note that we assume the probabilistic relational database as well as the result of every operation to be well-formed by applying the transformation rule in Equation 4.7.

140 Definition of selection ($\hat{\sigma}_p(R)$) A tuple r is only in the resultant relation in all worlds where it satisfies predicate p. Since the predicate p is a certain predicate, the descriptive sentence is not in any way impacted, and τ_σ is the identity function.

$$\varphi \overset{\tau_\sigma}{\mapsto} \varphi$$

$$\frac{r \in R \quad p(r)}{r \in \sigma_p(R)} \xrightarrow{\sigma} \frac{\begin{array}{c}\varphi' = \tau_\sigma(\varphi) \\ \langle r, \varphi\rangle \in R \quad p(r)\end{array}}{\langle r, \varphi'\rangle \in \hat{\sigma}_p(R)}$$

141 Definition of projection ($\hat{\pi}_{i_1..i_k}(R)$) A tuple projected tuple created from r is a member of the resultant relation in any world where it is a member of R. Therefore, the τ_π function is the identity function.

$$\varphi \overset{\tau_\pi}{\mapsto} \varphi$$

$$\frac{\begin{array}{c} r \in R(at_1, \ldots, at_n) \\ \{i_1, \ldots, i_k\} \in 1..n \end{array}}{\langle r.at_{i_1}, \ldots, r.at_{i_k}\rangle \in \pi_{i_1..i_k}(R)} \xrightarrow{\pi} \frac{\begin{array}{c} \varphi' = \tau_\pi(\varphi) \\ \langle r, \varphi\rangle \in R(at_1, \ldots, at_n) \\ \{i_1, \ldots, i_k\} \in 1..n \end{array}}{\langle\langle r.at_{i_1}, \ldots, r.at_{i_k}\rangle, \varphi'\rangle \in \hat{\pi}_{i_1..i_k}(R)}$$

142 Definition of cartesian product ($R \hat{\times} S$) A tuple rs is a member of the resultant relation in any world where the tuples r and s are members of R and S, respectively. The $\tau_\times$ function expresses this dependency by a conjunction of the tuples' descriptive sentences.

$$\varphi, \psi \overset{\tau_\times}{\mapsto} \varphi \wedge \psi$$

$$\frac{r \in R \quad s \in S}{rs \in R \times S} \xrightarrow{\times} \frac{\begin{array}{c}\varphi' = \tau_\times(\varphi, \psi) \\ \langle r, \varphi \rangle \in R \quad \langle s, \psi \rangle \in S\end{array}}{\langle rs, \varphi' \rangle \in R \hat{\times} S}$$

143 Definition of join ($R \hat{\bowtie}_p S$) The join operation can be expressed as $\bowtie_p = (\sigma_p \circ \times)$, a composition of the product and selection operations. This composition holds true in probabilistic relation algebra as well. The $\hat{\bowtie}_p$ operation can be expressed as $\hat{\bowtie}_p \equiv (\hat{\sigma}_p \circ \hat{\times})$. For the sake of completeness we present the derived full definition below.

$$\varphi, \psi \overset{\tau_\bowtie}{\mapsto} \varphi \wedge \psi$$

$$\frac{r \in R \quad s \in S \quad p(rs)}{rs \in R \bowtie_p S} \xrightarrow{\bowtie} \frac{\begin{array}{c}\varphi' = \tau_\bowtie(\varphi, \psi) \\ \langle r, \varphi \rangle \in R \quad \langle s, \psi \rangle \in S \quad p(rs)\end{array}}{\langle rs, \varphi' \rangle \in R \hat{\bowtie}_p S}$$

A tuple rs is a member of the resultant relation in any world where where it is a member of both R and S, and satisfies the predicate p. The predicate p is a certain predicate, and has no impact on the possible worlds in which rs holds. Therefore, $\tau_\bowtie$ only needs to express the dependency of rs on the presence of r and s in R and S respectively.

144 Definition of union ($R \hat{\cup} S$) A tuple r or s is in the resultant relation in all worlds where it is also in one or both of the operand relations. Therefore,

the $\tau_\cup$ function is the identity function.

$$\varphi \overset{\tau_\cup}{\mapsto} \varphi \qquad\qquad \varphi \overset{\tau_\cup}{\mapsto} \varphi$$

$$\frac{r \in R}{r \in R \cup S} \quad \frac{s \in S}{s \in R \cup S} \quad \xrightarrow{\cup} \quad \frac{\begin{array}{c}\varphi' = \tau_\cup(\varphi) \\ \langle r, \varphi\rangle \in R\end{array}}{\langle r, \varphi'\rangle \in R \hat{\cup} S} \quad \frac{\begin{array}{c}\psi' = \tau_\cup(\psi) \\ \langle s, \psi\rangle \in S\end{array}}{\langle s, \psi'\rangle \in R \hat{\cup} S}$$

145 Definition of intersection ($R \hat{\cap} S$) A tuple r is in the resultant relation in all worlds where it is also in both the operand relations. Therefore, the $\tau_\cap$ function is the conjunction of descriptive sentences, expressing the dependency that r must be a member of both operands.

$$\varphi, \psi \overset{\tau_\cap}{\mapsto} \varphi \wedge \psi$$

$$\frac{r \in R \quad r \in S}{r \in R \cap S} \quad \xrightarrow{\cap} \quad \frac{\begin{array}{c}\varphi' = \tau_\cap(\varphi, \psi) \\ \langle r, \varphi\rangle \in R \quad \langle r, \psi\rangle \in S\end{array}}{\langle r, \varphi'\rangle \in R \hat{\cap} S}$$

146 Definition of difference ($R \hat{\setminus} S$) A tuple r is in the resultant in worlds where it is a member of R but not of S. This leads to a two part definition.

$$\varphi \overset{\tau_\setminus}{\mapsto} \varphi \qquad\qquad \varphi, \psi \overset{\tau_\setminus}{\mapsto} \varphi \wedge \neg\psi$$

$$\frac{r \in R \quad r \notin S}{r \in R \setminus S} \quad \xrightarrow{\setminus} \quad \frac{\begin{array}{c}\varphi' = \tau_\setminus(\varphi) \\ \langle r, \varphi\rangle \in R \quad r \notin A(S)\end{array}}{\langle r, \varphi'\rangle \in R \hat{\setminus} S} \quad \frac{\begin{array}{c}\varphi' = \tau_\setminus(\varphi, \psi) \\ \langle r, \varphi\rangle \in R \quad \langle r, \psi\rangle \in S\end{array}}{\langle r, \varphi'\rangle \in R \hat{\setminus} S}$$

If there are worlds where $r \in R$ yet there is no world where $r \in S$ this leads to the first definition. Here r is a member of the resultant relation in all worlds where it is in R. In this case $\tau_\setminus$ is the identity function.

Alternatively, if there are some worlds where r is also in S this leads to the

second definition. Here r can only by in the resultant relation in worlds where it is in R but not in S, which is expressed by the two-argument $\tau_\setminus$ function.

5.3.2 Named Entity Extraction and Disambiguation

The natural language case presented in Section 1.7.1 provides the ambiguous sentence "Paris Hilton stayed in the Paris Hilton". In this example we show how a probabilistic relational model can help determine contextual information such as the country of locations. In a next processing step, this established context could be used to reason about the likelihood of a certain interpretation.

Figure 5.7 contains an example of the application of probabilistic relational algebra for our running example. Relation Type is an excerpt of Figure 1.2. Using relations RefersTo and Gazetteer we compute a new relation Locations with possible countries for the named entities:

$$\hat{\pi}_{\mathsf{phrase,pos,country}}(\hat{\sigma}_p(\mathsf{Type}\,\hat{\times}\,\mathsf{RefersTo}\,\hat{\times}\,\mathsf{Gazetteer})) \tag{5.26}$$

$$\begin{aligned} \text{where } p = (&\mathsf{Type.phrase} = \mathsf{RefersTo.phrase} \\ &\wedge\, \mathsf{Type.pos} = \mathsf{RefersTo.pos} \\ &\wedge\, \mathsf{RefersTo.gazetteer} = \mathsf{Gazetteer.id}). \end{aligned} \tag{5.27}$$

The resulting relation Locations gives two possible alternatives for the country of the location indicated with "Paris" in position 1. Both require that "Paris" actually refer to a place through the y=1 label.

5.3.3 Restrictions and MayBMS

The probabilistic relational model behind MayBMS uses similar notions as our framework:

- MayBMS's *random variable* (RV) is similar to our partitioning ω.
- MayBMS's *random variable assignment* (RVA) is similar to our label ω=v.

Type

phrase	pos	type	φ
Paris Hilton	1,2	person	x=1
Paris Hilton	1,2	hotel	x=2
Paris	1	place	y=1
Hilton	2	brand	z=1

Gazetteer

id	spelling	country	φ
g11	Paris	France	$\top$
g12	Paris	Canada	$\top$

RefersTo

phrase	pos	gazetteer	φ
Paris	1	g11	a=1
Paris	1	g12	a=2

Locations

phrase	pos	country	φ
Paris	1	France	$y{=}1 \wedge a{=}1 \wedge \top$
Paris	1	Canada	$y{=}1 \wedge a{=}2 \wedge \top$

Figure 5.7: Example relations with descriptive sentences. The 'Locations' relation is the result of $\hat{\pi}_{\mathsf{phrase,pos,country}}(\hat{\sigma}_p(\mathsf{Type} \hat{\times} \mathsf{RefersTo} \hat{\times} \mathsf{Gazetteer}))$

- MayBMS's *world set descriptor* (WSD) is similar to our descriptive sentence φ.
- MayBMS's *world set* (WS) is similar to our set of introduced partitionings Ω including attached probabilities $P(\omega{=}v)$.

The probabilistic relational model we created by applying our framework has the full expressive power of the framework behind it. A means for gaining performance is to impose restrictions on the expressivity and to omit operations. For example, the omission of the difference operation $\hat{\setminus}$ creates a situation where negation is no longer introduced into descriptive sentences.

Instead of implementing our own probabilistic relational DBMS, we will show how MayBMS is a restricted variant of our probabilistic relational model.

147 Restricted descriptive sentences The most important restriction that MayBMS imposes is that it allows only conjunctive sentences. This makes

their representation in the implementation much easier, since they can no longer form arbitrarily complex nested expressions.

Additionally, by only allowing conjunctive combination of two descriptive sentences we guarantee that any such combination of two sentences describes an equal or smaller set of possible worlds:

$$W(\varphi \wedge \psi) \subseteq W(\varphi) \tag{5.28}$$

$$W(\varphi \wedge \psi) \subseteq W(\psi) \tag{5.29}$$

In the implementation this leads to the situation where, due to the knowledge that we always reduce the set of possible worlds by combining sentences, we can discard any answer for which both sentences contain a label from the same partitioning for different partitions. For example, if the combined sentence $\varphi = (x{=}1 \wedge y{=}2) \wedge (y{=}1)$ occurs, we know that $W(\varphi) = \varnothing$ and we can discard the answer immediately.

148 Non well-formed answers Restricting descriptive sentences to conjunctions severely limits the ability to express the results of operations. For example, the union operation as defined in Paragraph 144 is no longer possible. If a tuple r is in both R and S with different descriptive sentences, the well-formedness rule of Equation 4.7 introduces a disjunction. The same holds for many other operations that inadvertently introduce non well-formed tuples into the resultant relation.

MayBMS copes with this by not enforcing well-formedness. By letting go of well-formedness it becomes possible to continue reasoning with only conjunctive sentences. When a single sentence is necessary, the disjunctive normal form can quickly be determined by collecting all descriptive tuples that annotate the same assertion:

$$\varphi_{\mathrm{DNF}}(a) = \bigvee_{\langle a,\psi\rangle \in R} \psi \tag{5.30}$$

In MayBMS, the construction of the disjunctive normal form is omitted and the set $\{\psi \mid \langle a, \psi\rangle \in R\}$ is used directly for probability calculations and estimates.

In Chapter 6 we will validate the application of this restricted probabilistic relational model as used by MayBMS to combine grouping information from three different sources.

5.4 Conclusions

In this chapter we presented probabilistic variants of Datalog (Section 5.1), XPath (Section 5.2), and SQL (Section 5.3). Each of these variants was created by applying our framework from Chapter 4. The ease with which we are able to turn these certain data models into probabilistic variants shows the data independence of our framework.

The framework gives rise to two broad categories of optimisations. The first category includes sentence manipulations, and is purely based on the properties and laws of propositional logic. The second category of optimisations are those that make use of the implications of the underlying data model. This distinction indicates that it is possible to abstract certain optimisations into a a generic uncertainty management component that can be applied regardless of data model.

The framework's application to both Datalog and XPath creates very expressive probabilistic variants of both. As we will further investigate in Chapter 6 for the relational model, the created systems are robust enough to be used in real-life situations. The JudgeD system has been used in maritime evidence combination [46], and is also being used in an educational context.

CHAPTER 6

Case: Homology Integration

Parts of this chapter have been published as [116].

In the previous chapter we showed that our framework from Chapter 4 is data model independent by applying it to three data models. In this chapter we continue our investigation of the practical qualities of the probabilistic relational model, as defined in Section 5.3, with a probabilistic approach for integrating data on a bioinformatics use case concerning homology, as presented in Section 1.7.3. Homology data consists of groupings of proteins that are expected to have the same function in different species. A bioinformatician has a large number of homology data sources to choose from. To enable querying combined knowledge contained in these sources, i.e., what 'science knows so far', they need to be integrated. We validate our method of Chapter 2 by integrating three real-world biological databases on homology in three iterations of refinement. The fact that attempts at a similar integration are made in the field itself, e.g. [71], illustrates that this is a representative case.

6.1 Introduction

As explained in Chapter 2, probabilistic approaches for data integration have much potential [75] for improving integration quality. We view data integration as an iterative process where data understanding gradually increases as the data scientist continuously refines his view on how to deal with learned intricacies

like data conflicts.

149 Repurposing bioinformatics data sources An important part of the field of bioinformatics is about combining available data sources in novel ways in a pursuit to answer new, far-reaching research questions. A bioinformatician typically has a large number of data sources to choose from, created and cultivated by different research institutes. Some are curated or partially curated, while others are automatically generated.

Though bioinformaticians are knowledgeable in the field and aware of the different data sources at their disposal and methods used, they do not know the exact intricacies of each data source. Therefore, a bioinformatician typically obtains a desired integrated data set not in one attempt, but after several iterations of refinement.

Most data sources are created for a specific purpose. A bioinformatician's use typically goes beyond this foreseen use. The act of repurposing of the data, i.e., using the data for a purpose other than its intended purpose, is another source of integration complexity. For example, the quality of data in a certain attribute may be lower than required.

In short, data understanding is a continuous process, with the bioinformatician's understanding of the intricacies of data sources growing over time. It is therefore required that this evolving knowledge can be expressed and refined. We call this specification an *integration view*. Querying and analysing the result of a refined integration view produces more understanding which is in turn used to further refine the integration view.

150 ProGMAP, a specialised homology tool [71] presents the tool ProGMAP for the comparison of orthologous protein groups from different databases. Instead of integrating protein groups, ProGMAP assists the user in comparing protein groups by providing statistical insight. Groups are compared pairwise and various visual display methods assist the user in assessing the strengths and weaknesses of each database.

Our approach differs from ProGMAP in that we want to provide the user

with a technique to query the combined data sources, instead of assisting the user in comparing them. In essence, homology data represents groups of proteins that are expected to have the same function in different species. Obtained by using different methods, the sources only partially agree on the homological relationships. Combining them allows for querying and analysing the combined knowledge on homology.

151 Problem statement We generalise the homology case by viewing it as the problem of integrating data on groupings. We define a *data source* S_i as a database containing elements D^i_E and groups D^i_G where:

$$\forall g \in \mathrm{D}^i_G : g \subseteq \mathrm{D}^i_E \tag{6.1}$$

Each source holds information on different sets of proteins, i.e., the various D^i_E partially overlap. The goal is to construct a new data set with groups over $\bigcup_i \mathrm{D}^i_E$ that allows for scalable querying for questions like "Which elements are in a group with e?" and "Are elements e_1 and e_2 in the same group?".

152 Global approach We focus on an iterative probabilistic integration of the grouping data. It is based on the generic probabilistic data integration approach of [60] which constructs a *probabilistic database*. We call this representation an *uncertain grouping*. Being probabilistic, the above queries return possible answers with their likelihoods.

An uncertain grouping is a grouping of elements for which the true grouping is unknown, but which faithfully represents the user's critical and fine-grained view on how much the data elements and query results can be trusted. Although probabilistic data integration is an active research problem [75], there is to our knowledge no work on probabilistic integration of data on groups.

153 Applying continuous refinement The process of continuous combination, as described in Chapter 2, is iterative. It starts out with a very simple set of combination rules. In the case of combining groupings, a good start is

a simple *integration view* such as 'one-database-source-is-entirely-correct-but-it-is-unclear-which-one'. One naturally discovers the limitations of this view while using the resulting data.

Subsequently, more fine-grained integration rules are specified which combine the data in a better way, deals with conflicting data in a better way, and specifies better likelihoods for certain portions of the data to be correct (trust assignment). The integration view allows for an automatic re-construction of the integration result. As long as the integration result is not good enough, the process is repeated leading to handling inconsistencies and ambiguities at ever finer levels of granularity.

154 Generalised application of combining groupings The technique we propose works for categorisations and groupings of items. Such groupings are often encountered in data sources. They originate from automatic classifiers such as machine learning or data mining approaches, but also from human experts. Such data sources are not guaranteed to be correct. Measurement errors, data entry errors, or predictive heuristics may produce partially incorrect data.

For example, an administration of project teams may be incorrect if it can not keep up with people moving from team to team, get ill for possibly longer periods, etc. A solution direction for higher data quality here, would be to combine the administration with other independent data sources or other methods for determining team membership. For example, company-wide software for cooperative work (discussion boards, task boards, etc.) may be used to extract an apparent cooperation, hence team membership.

Another example is the classification of scientific articles. Libraries typically use both manual as well as automatic classification mechanisms. The correctness of the resulting classifications are affected by either the judgement of human classifiers or by the applied automatic keyword clustering algorithms. By combining multiple sources of article classifications (curated indices, automatic keyword clustering results, etc.), one may improve the overall quality of the classification.

155 Contributions and Outlook We present a technique for combining grouping data from multiple sources. The main contributions of this chapter are:

- A generic probabilistic approach to combining grouping data in which an evolving view on integration can be iteratively refined.
- Validation of the method of Chapter 2 for iterative refinement in small steps enabled by probabilistic database technology on a real-world data integration case.
- Experimental evaluation of the maturity of probabilistic relational database technology for a real-world probabilistic data integration case with sizeable data volumes.

The homology case is explained in Section 1.7.3. We generalise the homology case to the problem of integrating grouping data and elaborate on how our probabilistic integration approach addresses this problem.

The rest of this chapter is laid out as follows: Section 6.2 illustrates our method of Chapter 2 by presenting an iterative probabilistic integration of three data sources with data on homology. Each iterative produces an integration view that can be meaningfully used for querying and analysis, which is then further refined based on the obtained increase in data understanding. Section 6.3 discusses the flexibility of iterative refinement of integration views. Section 6.4 describes an evaluation of applying the method of Chapter 2 and framework of Chapter 4 to the real-world case using an existing probabilistic database the conforms with our framework. The evaluation is performed quantitatively by carrying out experiments that measure performance and scalability, as well as qualitatively by discussing how well the implementation as well as the probabilistic framework functioned in accomplishing the integration and querying tasks of the case. Section 6.5 discusses, among other things, the complexity of the use case and the scalability of our technique. We conclude the chapter in Section 6.6.

Data sources			
S_1 :	ABC_1	DE_1	FG_1
S_2 :	AB_2	CD_2	FH_2
S_3 :		ABE_3	FGH_3

Legend	
S_i	Source i
XYZ_i	Group of 3 elements (from S_i)

Figure 6.1: Running example based on homology case in Section 1.7.3.

6.2 Iterative Integration Views

In this section we explain our iterative probabilistic integration approach in more detail. We start with a summary of the running example, followed by detail of the application of integration views. Finally, we discuss the extensibility of the integration view approach.

156 The Paperbird species Recall our fictitious Paperbird taxa from Section 1.7.3. The taxa features three species of Paperbird. Each of these three species evolved from the Ancient Paperbird, the extinct ancestor species of the paperbird genus. The Ancient Paperbird is conjectured to have genes K L M.

Two genes are orthologous if they have the same function in different species. For example, genes D and E are known to govern the length of the beak. Based on this, and on the conjectured function of the beak curvature function ancestor gene L, we call D and E orthologous, with L as common ancestor. If the algorithm used to construct source S_1 comes to the same conclusion, it will contain an orthologous group DE_1.

157 Paperbird homology data sources Figure 6.1 presents three example data sources, each containing two or three orthologous groups. We use the notation XYZ_i for a group of three elements, X, Y, and Z originating from source S_i. Observe that not every source is complete, for example, S_2 does not mention E. It depends on the source whether this absence means:

1. E is implicitly a group on its own,

2. E is does not belong to any group, or

3. it is unknown to which group E belongs.

This ambiguity is further discussed in Paragraph 175.

From Section 1.7.3, we know that in our fictitious reality the correct grouping is ABC, DE, and FGH. Observe that none of the sources in Figure 6.1 is complete and fully correct. A bioinformatician integrating these sources, however, does not know what is the correct grouping, not even how well he can trust the data. The goal is to determine based on current scientific knowledge contained in the sources, what the correct grouping is, or rather, the confidence in possible groupings.

158 Creating alternatives from uncertain groupings We model an uncertain grouping as a probabilistic database adhering to the possible worlds model. In this model, an uncertain grouping is a compact representation of many possible groupings: the possible worlds. Probabilistic database technology is known to allow for scalable querying of an exponentially growing number of possible worlds [5]. Querying in a possible worlds model means that the query result is equivalent with evaluating the query on each possible world individually and combining those answers into one probabilistic answer.

Although we abstract from what an *integration view* exactly looks like, one can regard it as a set of data integration rules. These rules specify not only how the raw data should be merged, but also which relevant alternatives exist in case of conflicts as well as what confidence to assign to certain portions of the data and such alternatives.

159 Introducing the integration views Our method of working with integration views is iterative, i.e., one starts with a simple view on how the data should be integrated and trusted based on initial assumptions that may or may not be correct. By evaluating and using the integrated result, a bioinformatician gains more understanding in the data, which (s)he uses to adapt and refine the integration view.

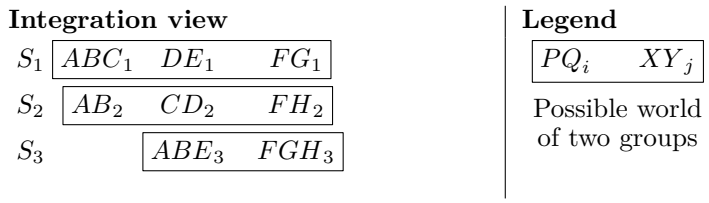


Figure 6.2: Depiction of integration view SRC: each source is a possible world, which ultimately produces 3 worlds.

The reason behind this way of working is, that we believe, as we stated before, that data understanding is a continuous process, with the bioinformatician's understanding of the intricacies of each data source growing over time. With the integration view method, the bioinformatician is able to express and refine his evolving opinion on the reliability of the data in the sources and how the data should be combined. He can then query and analyse the result of his actions to see how they reflect on the results.

In the sequel, we illustrate the method by going through three iterations, each centred around a different integration view and evaluate the evolving integrated data.

160 The SRC integration view Suppose we would start with taking the simplistic view of 'one-data-source-is-entirely-correct', SRC for short: the belief that one source is entirely correct, but it is unknown which one. In this view, each data source is a possible world (see Figure 6.2). There is basically one *choice*: which alternative data source is the correct one: S_1, S_2, or S_3.

Although simple, this rule does provide a meaningful integration that can immediately be used for querying and analysis. For example, a probabilistic database containing the data integration result will be able to directly answer the question "which proteins are homologous with A". The answer will be:

- B for certain (since all sources agree on A and B being homologous).
- Possibly C (since source S_1 claims it).

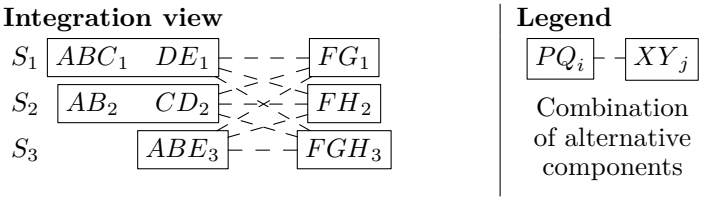


Figure 6.3: Depiction of integration view COMP: a possible world is a combination of independent components, which ultimately produces 9 worlds.

- Possibly B and E as well (since source S_3 claims it).

Nevertheless one may discover during exploration or analysis, that the integration is inadequate. A typical phenomenon of improved data understanding is that one questions one's assumptions that underly the integration view. As a consequence, other more fine-grained views are defined on combining the data in the sources, which typically leads to more choices.

161 The COMP integration view Suppose that in this case we question our simplistic integration view: because it states that 'one-data-source-is-entirely-correct', it precludes that one source may be correct for some groups of proteins and another source is correct for other groups of proteins. As a resolution of this one observation, one could argue that the disputes among the sources around elements A, B, C, D, E and around F, G, H are independent of each other, hence that, say, S_1 could be correct on the component A, B, C, D, E and S_2 on F, G, H. In this view, the combination $\{ABC_1, DE_1, FH_2\}$ should be among the possible worlds (see Figure 6.3).

The general rule of this view, COMP for short, is that the independent *component*s of groups under dispute, can be freely combined to form possible worlds. In the example, the view results in two independent choices, one for each component, with each three alternatives resulting in $3 \times 3 = 9$ possible worlds.

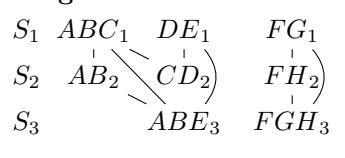


Figure 6.4: Depiction of integration view COLL: a possible world is a collision-free combination of groups which ultimately produces 2^9 worlds.

162 The COLL integration view To illustrate the flexibility of our approach, we present a third even more fine-grained collision-based integration view, called COLL. Two groups *collide* iff they overlap but are not equal. Figure 6.4 shows the collisions between groups in our example.

The idea behind the COLL-view is that if two sources disagree on a group, i.e., the groups collide, only one can be correct (actually, this is a simplification as both can be incorrect; see Section 6.5) In other words, each collision is in essence a choice. Note, however, that there are dependencies between these choices. For example, consider collisions ABC_1–AB_2 and DE_1–CD_2. If they were independent, then $2 \times 2 = 4$ combinations of groups would be possible, but the combination $\{ABC_1, CD_2\}$ violates the important grouping property that each element can only be a member of one group. Therefore, the general rule for this integration view is that all *collision-free* combinations of groups form the possible worlds.

One can see that the COLL method is more fine-grained by observing that $\{ABE_3, CD_2, FG_1\}$ is a possible world that is not considered by SRC nor COMP. Without any dependencies, n binary choices would generate 2^n possible worlds. In the example, the view would result in $2^9 = 512$ worlds if there would be no dependencies. With dependencies, the number of possible worlds in the example is reduced to 40.

Note that this includes the empty world, because in the probabilistic database, every fully described sentence $\bar{\varphi}$ that describes a world with one or more collisions, actually describes a world without data, i.e., an empty world (see Paragraph 164 for more details).

D

	group	φ
d_1	ABC_1	$r_1{=}1 \wedge r_2{=}1 \wedge r_3{=}1$
d_2	DE_1	$r_5{=}1 \wedge r_6{=}1$
d_3	FG_1	$r_7{=}1 \wedge r_8{=}1$
d_4	AB_2	$r_1{=}2 \wedge r_4{=}1$
d_5	CD_2	$r_2{=}2 \wedge r_5{=}1$
d_6	FH_2	$r_7{=}2 \wedge r_9{=}1$
d_7	ABE_3	$r_3{=}2 \wedge r_4{=}2 \wedge r_6{=}2$
d_8	FGH_3	$r_8{=}2 \wedge r_9{=}2$

Ω

l	P	
$r_1{=}1$	p_1	'S_1 correct' for ABC_1–AB_2
$r_1{=}2$	p_2	'S_2 correct' for ABC_1–AB_2
$r_2{=}1$	p_3	'S_1 correct' for ABC_1–CD_2
$r_2{=}2$	p_4	'S_2 correct' for ABC_1–CD_2
⋮		
$r_8{=}1$	p_{15}	'S_1 correct' for FG_1–FGH_3
$r_8{=}2$	p_{16}	'S_3 correct' for FG_1–FGH_3
$r_9{=}1$	p_{17}	'S_2 correct' for FH_2–FGH_3
$r_9{=}2$	p_{18}	'S_3 correct' for FH_2–FGH_3

Figure 6.5: Probabilistic database representation $CPDB = (D, \Omega)$ for the uncertain grouping constructed under integration view COLL (see Figure 6.4).

6.3 Flexibility of Integration Views

Typically one would have many more considerations, sometimes rather fine-grained, that one would like to 'add' to one's integration view. For example, a bioinformatician may believe that groups CD_2 and FH_2 are extra untrustworthy, because he holds the opinion that the research group who determined those results is rather sloppy in the execution of their experiments. Or, he may have more trust in curated data, or even different levels of trust for data curated by different people or committees. Our approach can incorporate such considerations as well by adding partitionings which effectively add worlds in which the data does not exist, hence as a consequence reduce the probability that precisely those data items are true.

We argue that integration problems such as conflicts, ambiguity, trust, etc. can all be modelled in terms of choices that can be formalised with random events, which in turn can be represented in a probabilistic database with random variables (partitionings) and annotating tuples with world set descriptors (sentences) composed of random variable assignments (labels). In this section, we like to emphasise the flexibility of the approach.

163 Nuanced relations through dependencies Consider for example the probabilistic database constructed for the paperbird example according to integration view COLL as illustrated in Figure 6.5. Observe how the 9 collisions result in 9 random variables (partitionings) in a straightforward way. Furthermore, the concept of collision-freeness is represented in the world set descriptors (sentences). For example, tuple ABC_1 can only exist if all collisions in which it is involved fall in its favour.

Observe also how such an intricate integration view as COLL, does not produce more tuples in the **group** table, only the world set (set of partitionings) grows because of the higher number of choices, and the world set descriptors (sentences) become larger because of the need to faithfully represent the dependencies between the existence of tuples caused by the collision-freeness condition.

Nevertheless, this is only more data. We show in Section 6.4 that this does not cause scalability problems even in a voluminous real-world case such as homology.

164 Taking into account empty worlds As described in Paragraph 101 there is the possibility of answers only holding in worlds that are impossible due to having an inconsistent world set descriptor (sentence), answers only holding in worlds that have a probability of zero. In the case of COLL the opposite is the case.

The possible answers to a query come with a probability for the trustworthiness of the answer, essentially the combined probability of all worlds that agree on that answer. Note that our modelling of COLL induces empty databases for world set descriptors (sentences) that would lead to one or more collisions. One could normalise the probabilities of query answers with $1 - \mathrm{P}(\varnothing)$, the combined probability of all collision-free combinations.

165 Iterative refinement of the integration view Finally, we would like to emphasise that the process of discovering integration issues and imposing the associated consideration on the data by refining one's integration view, is

an iterative process. We claim that such considerations can be imposed on the data by introducing more random variables (partitionings) and adding RVAs (labels) to the WSDs (sentences) of the appropriate tuples.

Recall, for example, the issue of the sloppy research group at the start of this section. Here, one new random variable (partitioning) can be introduced and a RVA (label) added to the WSD (sentence) of all tuples of this research group. After such a refinement, the bioinformatician obtains a database that can be directly queried so that he can examine its consequences. He thus iteratively refines his integration view until the data faithfully expresses his opinions as well as the result of any query or analysis run on this data.

6.4 Evaluation

Two main questions guide the evaluation: can our method of Chapter 2 be applied in a realistic real-world case using an existing probabilistic database conforming with our framework of Chapter 4? And if so, how well does it scale to realistic amounts of data? In particular to determine if current probabilistic database technology can cope with the amounts of uncertainty introduced by our framework. The second question is addressed quantitatively by conducting experiments and measuring performance and scalability. The first question is addressed qualitatively by discussing how well the implementation as well as the probabilistic framework functioned in accomplishing the integration and querying tasks of the case.

166 Test set construction For the evaluation, we constructed a test set of homology data from the biological databases Homologene (release 67, [84]), PIRSF (release "2012_03", [120]), and eggNOG (release 3.0, [91]). The groupings from each were loaded into a single database for the construction of the integration views and querying. Where necessary database-specific accession numbers were converted to UniProt accession numbers. This ensures that identical proteins in different groups are correctly referenced.

167 Query suites Two query classes can be distinguished among queries commonly executed on homology databases:

1. **single**: "Which proteins are homologous with X?" with X a known protein.

2. **pair**: "Are X and Y homologues?" with X and Y known proteins.

Based on these two query classes we generate query suites based on sampling proteins from the combined database. These two suites are each generated with a specific purpose in mind:

1. 1000 single and 1000 pair queries. All pairs are guaranteed to have a homologous relation. This suite is used to determine average query execution times for all integration views.

2. 100 single queries and 200 pair queries. For the latter, 100 queries have a homologous relation and the other 100 do not.

The random variable assignments (labels) for the integration views SRC, COMP and COLL were generated according to our integration views as described in Section 6.2. Since actual probabilities do not influence performance, probabilities were assigned uniformly over the RVAs.

To study scalability regarding amounts of uncertainty, the WSD (sentence) size is used as an artificial bound on the amount of uncertainty. Both SRC and COMP feature only a single RVA (label) hence are effectively equivalent with respect to execution time. Due to technical limitations (see Section 6.4.2), the integration view COLL was produced with a maximum of 500 RVAs (labels) per WSD (sentence). This did not hinder the experiments, because we were interested in studying the execution behaviour by varying the WSD (sentence) size anyway. To this end, integration views COLLN ($N = 500, 450, \ldots, 100, 50$) were generated by truncating the WSD (sentence) of all tuples to size N. Hence, N is the maximum length of the WSDs (sentences) (the majority of the WSDs indeed is exactly of size N). No size indication means COLL500.

Integration view	Mean query time (ms)	Standard deviation
SRC	18.627	26.864
COMP	19.061	27.569
COLL	23488.197	93184.375

Table 6.1: Mean query times for query suite 1.

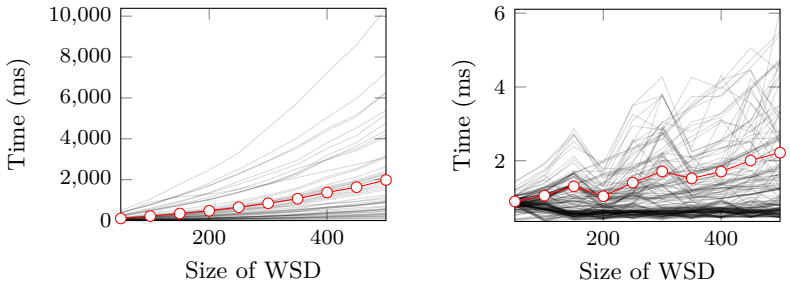
168 Probabilistic database We use the probabilistic database system MayBMS [5]. Because of experimenting with an existing system, we accept some technical limitations inherent in these systems. Completely overcoming these limitations is not the focus of our evaluation. A note on these limitations can be found in Section 6.4.2.

6.4.1 Experiments

The following three experiments investigate how well our approach scales for realistic amounts of data. The experiments were conducted on an Intel i7 x86-64bit with 7.7GB ram running Linux 3.2.0. Compilation was done with gcc 4.6.3.

169 Experiment 1: Mean query times Based on query suite 1, each query is executed 10 times. Mean query time per integration view is calculated from the latter 9 measurements; the first is discarded to prevent adverse effects of a 'cold' database. Table 6.1 presents preliminary results that show that the amount of uncertainty of each integration view has a large impact on the mean execution time. Large standard deviations indicate large variations of query times within each integration view. The following experiments investigate the cause of this variation.

170 Experiment 2: world set descriptor (sentence) size The goal here is to determine the impact of WSD (sentence) size on query execution time. Query suite 2 is used on integration views COLL50, COLL100, ..., COLL500.



(a) Mean query times per 'single' query. (b) Mean query times per 'pair' query.

Figure 6.6: Mean query time (in white-red) and distinct query times (in grey).

The experiment proceeds as follows: each query in the query suite is repeated 10 times, the first measurement is discarded. The mean query time per query are calculated based on the 9 time measurements.

Figure 6.6 presents the trend in mean query time with growing WSD (sentence) for both query classes separately. The 'pair' queries are orders of magnitude faster than the 'single' queries due to smaller amounts of uncertainty per query result. The two drops in Figure 6.6b at COLL200 and COLL350 are most likely due to favourable alignment of data in memory.

171 Experiment 3: Numbers of WSDs (sentences) and RVAs (labels) The goal here is to investigate the impact of the number of WSDs (sentences) and RVAs (labels) involved in answering a query on the query time. Query suite 2 is used on integration views COLL50, ..., COLL500.

A counting function is used to count the number of WSDs (sentences) used to answer the query, and the number of unique RVAs (labels) that were encountered while answering the the query. The counting function is applied to all queries from the 'single' and 'pair' suite for all trust views COLL50, ..., COLL500.

As can be seen in Figure 6.7 and Figure 6.8, the framework and MayBMS handle the real-world uncertainty well. For a large part, queries are executed within 2 seconds.

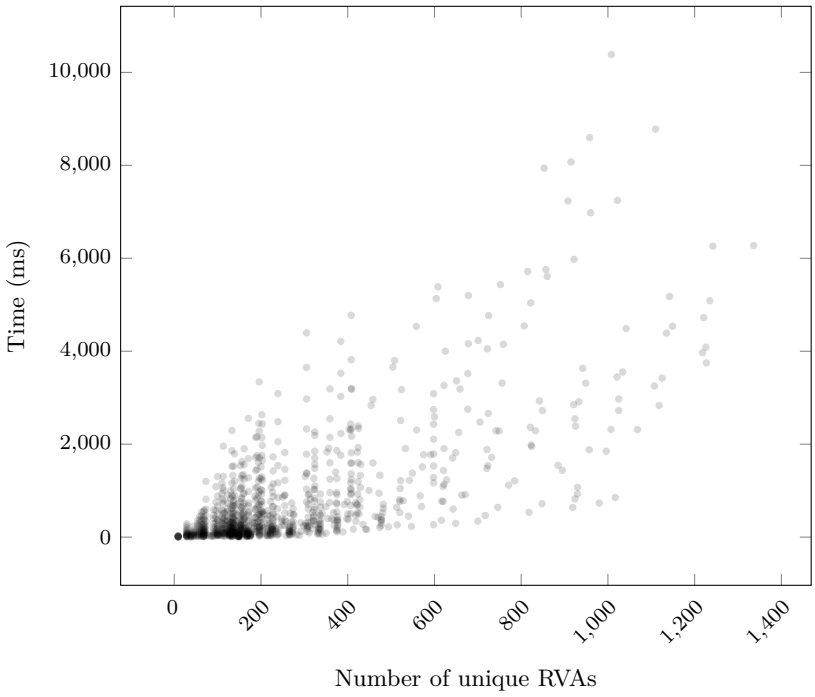


Figure 6.7: Impact of number of RVAs (labels) involved (all WSD (sentence) sizes; 'single').

The slower queries are slow due to a combination of a large number of unique RVAs (labels) and WSDs (sentences).

We conducted a further analysis of what execution time is spent on for the integration view with a large amount of uncertainty (COLL). Experiments with more data but similar amount of uncertainty show that data volume does not affect query execution times significantly. Furthermore, as Figure 6.8 shows, reddening of dots indicating longer query times is mostly found in the upper parts of the figure, i.e., with growing numbers of RVAs (labels) and not with higher numbers of WSDs (sentences). Based on these observations, we conclude that most time is consumed by confidence computation.

While conducting the experiments, a negligible number of queries did not

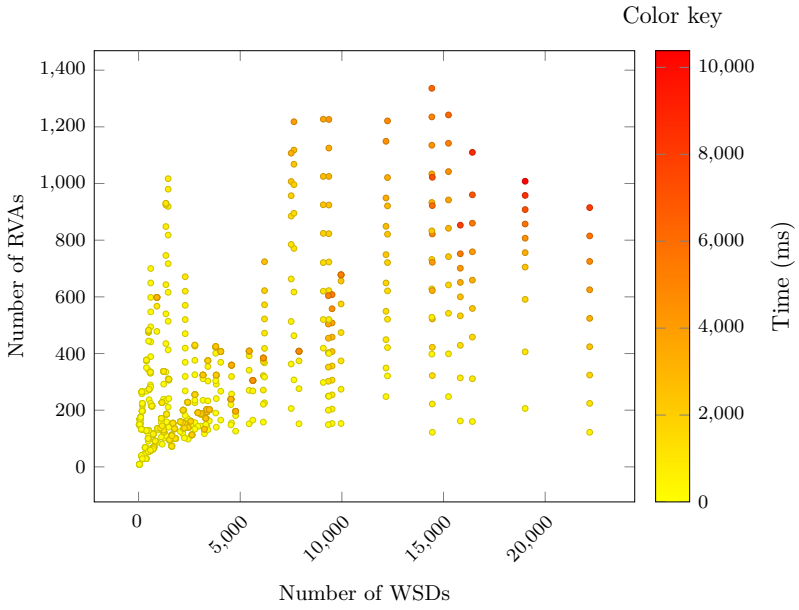


Figure 6.8: Impact of number of WSDs (sentences) and RVAs (labels) ('single').

finish. We suspect the method we use to interface with MayBMS to be the cause. Because our implementation is intended as a research prototype we have not spent significant effort on finding the cause, as it is not scientifically relevant.

6.4.2 Qualitative Evaluation

Next to evaluating the time performance of current probabilistic database technology, we have investigated the practical side of handling realistic amounts of uncertainty.

172 Technical limitations of MayBMS and PostgreSQL We ran into several technical limitations of PostgreSQL and MayBMS. According to the manual, and the source code, PostgreSQL tables are limited to 250–1600

columns, depending on column type. Since MayBMS uses a 3-column system for representing RVAs (labels), the limit on expressing RVAs (labels) is 83–533 per WSD (sentence) without actual data, and one less RVA (label) for each three columns of data. So, with 2 columns used up by other data, it can support at most 532 RVAs (labels) in the one WSD (sentence) associated with a tuple. Furthermore, MayBMS's confidence computation aggregates are implemented with stored procedures and PostgreSQL can not pass more than 100 arguments to a stored procedure. This limits the number of RVAs (labels) in the WSD (sentence) of a result tuple to 33.

To overcome the problem of not having more than 100 arguments to a function, we wrote our own representation of RVAs (labels) that is functionally equivalent to MayBMS' representation but allowed us to represent up to the limit of 532 RVAs (labels) in the WSD (sentence) of a tuple. We did so by taking advantage of the PostgreSQL ability to use arrays as a column type. By implementing a RVA base type, a WSD can be represented as an array of RVA values. Our implementation uses a custom aggregation function to feed our WSD representation to the MayBMS functions for confidence computation.

Conversion to our new representation can be done during integration view construction or querying. The impact of the overhead of conversion during querying was shown to be negligible. For the above experiments we converted during querying.

173 Further integration view refinement During analysis of the evaluation we encountered three measurements that qualify as outliers. Two of these outliers occurred for 'pair' queries with large execution times. As the experiments were conducted on a normal workstation, we strongly suspect that another program interfered with query execution.

One outlier occurred during the measurements of 'single' queries, specifically for protein F6ZHU6 (a UniProt identifier). This protein is related to muscle activity and is a member of an abnormally large number of orthologous groups. This outlier proves to be a valuable learning moment about the structure of orthologous groups. An unsuspecting bioinformatician himself would perhaps,

just like us, initially also assume that groups within one source are non-overlapping. For homology databases, one discovers that this is not true. According to bioinformatician A. Kuzniar whom we consulted about this issue:

> "The reason is that orthologous groups are nested as the orthology relations are defined based on a phylogenetic tree. Depending on how far you go back in time to infer these relations, e.g., for mammals (subset) vs. vertebrates (superset), there will be a different level of granularity in the orthologous groups. The overlap is between a superset and its subsets. However, things get more complicated when one also considers gene fusion events (hybrids) where two distinct genes in one species are fused together into a single gene in another species. In this instance, the tree model is inadequate and therefore one needs to resolve to a graph (network) model, see also [72]."

As the method set out in Chapter 2 aims to not spend effort if the current results are good enough, we have not attempted to resolve the issue. The current level of integration is good enough for our purposes.

The way the issue has been encountered in our own research is a nice illustration of data understanding being a continuous process that happens concurrently with the re-purposing, combination, and analysis of data from multiple sources. A next step in the refinement of the integration view could be the proper incorporation of this discovery.

6.5 Discussion

In this discussion we shortly touch on the subject of confidence approximation, followed by a discussion on the open world and closed world assumptions. We close with an interesting optimisation based on a graph representation of groupings.

174 Scalability and confidence precision The probabilistic data in our framework is composed of two parts. The first part is normal relational data, which scales as well as can be expected from a relational database. In the probabilistic integration of the grouping data, we do not generate additional normal data, so the amount of tuples is equivalent to the union of tuples from the data sources. All overhead, both in terms of space and computation time, is produced by the second part, i.e., the data representing the WSDs (sentences).

We currently use the exact confidence computation implemented by MayBMS and described in [66]. The COLL integration view generates one RVA (label) per collision. In this chapter we only took the first 500 collisions into account due to technical reasons. We have observed groups, however, that would generate a tuple with a WSD (sentence) composed of as much as 17885 RVAs (labels).

Because of this, the exact confidence computation has to deal with extremely small probabilities. Further work needs to be done to see whether approximate confidence computation, such as in [85], can be done over large amounts of RVAs (labels).

175 Open World versus Closed World Consider, for example, source S_1 and the fact that it does not mention H. Should this be interpreted (closed world assumption) as a statement that H is not orthologous to any other protein, in particular, F and G? Or (open world assumption) that S_1 does not make a statement at all about H, i.e., it may be orthologous to every protein?

Considering only sources S_1 and S_2 — note that S_2 does not mention G — one could hold the view that it is possible for G and H to be orthologous as both are possibly orthologous to F according to the respective sources. There is, however, no possible world in the uncertain grouping of S_1 and S_2 where G and H are in the same group using any of the integration view methods presented. Hence, the integration views of Section 6.2 all follow a closed world assumption.

The universe of discourse here is the domain of all proteins. Assuming that this domain is finite, one could theoretically construct an integration view

following an open world assumption by adding group tuples for all combinations of proteins and associating them with the appropriate WSDs (sentences). In practice, this is of course infeasible due to the sheer number of combinations. Below we present an attractive compromise between the open and close world assumptions which we coined the 'Tunnel vision' assumption.

176 Tunnel vision The idea of an open world can be applied in a restricted form: the world is assumed to be open only to the combined domain of the integrated sources, i.e., $\mathrm{D}_E^1 \cup \mathrm{D}_E^2$. We call this the tunnel vision world assumption as one does not view the world of the sources to be completely closed, also not completely open, but open/closed to the 'target world'.

In our example of combining S_1 and S_2, the combined domain of elements is $\mathrm{D}_E = \{A, \ldots, H\}$. A tunnel vision view can be achieved by adding possible group tuples to S_1 that include H and possible group tuples to S_2 that include E and/or G. Using either of the integration view methods, an uncertain grouping is established that includes the possibility that G and H are orthologous at the expense of a limited number of tuples and only one RVA (label) per unmentioned element per source.

Since the performance bottleneck of probabilistic databases does not reside in the query evaluation itself, but in the probability computation with growing WSDs (sentences), a tunnel-view is expected to be feasible in practice.

177 Graph representation and optimisation During our research, we explored alternative representations for groupings based on graph theory. The investigated graph-based representation is one in which each orthology relation is represented as an edge, and each protein as a vertex. Although a translation can be made from a groupings representation to a graph representation, the translation from graph representation to groupings representation was found to be problematic. Questions like "What other members are there in the groups containing protein X?" require clique-finding or a less precise form of clustering, which were found to be computationally undesirable. In the future, an investigation of power graph analysis [95] could provide new methods to

apply these optimizations.

This did lead us to an interesting venue for optimising the COLL integration view: if a set of collisions forms a clique, i.e., if all groups are mutually exclusive with each other, these dependencies can be expressed with a single random variable (partitioning). So any clique of n collision relations (which requires the introduction of n random variable and $2n$ random variable assignments) can be reduced to a single random variable and n random variable assignments.

This reduction does not change the semantics of the involved dependencies. It can be applied selectively on any number of cliques without creating an inconsistent state, allowing the optimisation to be executed incrementally during idle time.

6.6 Conclusions

Motivated by the real-world use case of homology, we propose a generic technique for combining groupings. Proteins in a homologous group are expected to have the same function in different species. Homology data is relevant when, e.g., a medicine is being developed and the potential for side-effects has to be determined. We combine 3 different biological databases containing homology data. We introduced this real-world use case of homology in Section 1.7.3.

In e-science as well as business analytics, data understanding is a continuous process with the analyst's understanding of the intricacies and quality of data sources growing over time. We propose a generic probabilistic approach to combining grouping data in which an evolving view on integration can be iteratively queried and refined. Such an 'integration view' models complications such as conflicts, ambiguity, and trust as probabilistic data.

Experiments show that our approach scales with existing probabilistic database technology. The evaluation is based on realistic amounts of data obtained from the combination of 3 biological databases, yielding 776 thousand groups with a total of 14 million members and 2.8 million random variables (partitionings).

Our technique allows a researcher (such as the bioinformatician) to focus

on the semantics of the data sources, instead of on the technical details of data integration. Integration choices can be modelled through the introduction of partitionings, instead of through directly changing the data itself, allowing the researcher to take a step back and look at the bigger picture, instead of worrying about each integration detail.

CHAPTER 7

Conclusions

Our global aim with the research presented in this thesis is to assist the process of repurposing data by developing generic technology assisting the process of data understanding and data combination. We phrased this goal with our problem statement "How to support scientists in understanding data semantics and data quality to speed up data intensive research?"

Achieving this goal requires overcoming the challenge of making explicit the quality of data and discovering the semantics of the reused data sources. Data quality is related to the original purpose of a data source. What is high quality for one purpose can be low quality for another purpose. Semantics, in so far that they can be communicated, are equally difficult to make explicit. To use a data source to its full potential requires understanding the semantics that source. Yet, the actual semantics of a data source are obscured by assumptions both on the producer and consumer side, and by data quality problems that lead to misunderstandings.

Additionally, the workflow of the e-scientist differs from the traditional business analytics workflow that works towards a single best integration of data sources. Science is about truth seeking and discovery, with this comes a more erratic workflow when compared to business analytics. This difference leads to a mismatch between current methods and tools based in business analytics and the data scientist's needs.

178 A method for data repurposing We have proposed an iterative method for data repurposing based on the principles of pay-as-you-go, good-is-good-enough and keep-track-of-your-stuff. This proposed method is an answer to our first research question:

RQ1 "What is a good method for data understanding, data repurposing, and data analysis?"

The proposed method is characterised by quickly iterating through the steps of analysis, exploration and feedback. After each iteration, the integrated data is in a usable state with unresolved integration issues being expressed as uncertainty in the data. Our method highlights opportunities where the domain expert can be assisted through tools and technologies. Several of these opportunities present themselves through the introduction of a personal knowledge base that contains the rules and choices built up by the domain expert over the course of data understanding and refining the integration.

We have looked at the practice of note taking through the lens of a traditional research laboratory. We highlighted the opposing desires of the scientist and the institute, and have investigated the compromise that has been established in such environments. Based on this compromise we sketched an answer to our second research question:

RQ2 "What tool support is a natural improvement of the documentation activities in an e-scientist's existing workflow?"

The three key features of freeform note taking, eventual observation of policy, and continuous policy evaluation form the basis of our approach to automated support for the well-established compromise. We presented the Strata system that implements the building blocks necessary to construct an automated lab notebook. The abilities of the system have been validated by prototyping a lab notebook system for the Prometheus laboratory.

179 A framework for probabilistic databases We revisited the formal foundations of probabilistic databases to answer our third research question:

RQ3 "What is a generic foundation for uncertain data management that fits the method of **RQ1**?"

We propose a formal framework that is based on attaching a propositional logic sentence to data assertions to describe the possible worlds in which that assertion holds. By doing so, the formalisation (a) abstracts from the underlying data model obtaining data model independence, and (b) separates metadata on uncertainty and probabilities from the raw data. In relation to the framework, we discuss open problems such as alternative data models, probability calculation, and aggregation, as well as scalability and optimisation issues brought to light due to the framework's properties.

We have validated the data model independence of the framework by applying it to Datalog, XPath and relational algebra to obtain probabilistic variants thereof. We have discussed the categories of optimisation that the framework presents, and how this impacts the implementation of a generic uncertainty management component. We have observed that the framework's application to both Datalog and XPath creates very expressive probabilistic variants of both, and that existing probabilistic relational database technology is equivalent to a restricted version of our probabilistic relational algebra.

We investigated the real-world use case of homology to answer our fourth research question:

RQ4 "How well can the foundation from **RQ3** be applied to a bioinformatics use case using existing probabilistic data management technology?"

Guided by the bioinformatics use case of homology we proposed a generic probabilistic approach to combining grouping data in which an evolving view on integration can be iteratively queried and refined. Such an 'integration view' models complications such as conflicts, ambiguity, and trust as probabilistic data. Experiments show that our approach scales with existing probabilistic database technology. The evaluation is based on realistic amounts of homology data obtained from the combination of 3 biological databases, yielding 776 thousand groups with a total of 14 million members and 2.8 million random variables (partitionings).

7.1 Released Software

In Paragraph 16 (Chapter 1) we stated our goal to place generated tools in the open source domain, and to ensure that these released tools have appeal beyond this thesis. This thesis has produced two released and fully implemented systems.

180 Strata: semi-structured data in a wiki In Chapter 3 we present the Strata system. Strata is built on the wiki system DokuWiki [28]. Within Strata it is possible to give a structured description of organisational constraints allowing automated assessment, to have multiple users collaborate on documenting their work, and to mix structured data and unstructured data. Strata's underlying data model is the well-known RDF, stored in a relational database management system. All triples in the relational database management system are derived from the structured data on the actual wiki pages. In effect, the RDBMS's function is to serve as an index to speed up query answering. At the time of writing the Strata system is used in at least 61 distinct wikis.

The strata system can be obtained from `https://github.com/bwanders/dokuwiki-strata/`. Further documentation and information can be found at `https://www.dokuwiki.org/plugin:strata`.

The Strata system has been released in the open source domain under the terms of the GPLv2 license. This permits commercial and private use, distribution and modification, all contingent upon the requirement that the source to modification is disclosed, and that all releases are under the GPLv2.

181 JudgeD: Probabilistic Datalog with dependencies In Chapter 5 we present the JudgeD system. JudgeD is an implementation of the probabilistic Datalog obtained by applying our framework from Chapter 4 to Datalog. In the JudgeD system one can express complex dependencies between arbitrary clauses, i.e., both facts and rules.

The JudgeD implementation can connect to external data sources through native predicates, and supports negative Datalog based on SLG resolution.

We have implemented a Monte Carlo approximation for calculating answer probabilities, and presented an exact solver that works for positive Datalog. The JudgeD system has been used in the real-world use case of maritime evidence combination [46], and sees use in an educational context.

The JudgeD system, manual and other documentation can be obtained from `https://github.com/utdb/judged`.

The JudgeD system has been released under the terms of the MIT license. This permits commercial and private use, distribution, and modification, all contingent on releasing modifications under the same license if they are released.

7.2 Future Work

At the core of the iterative method for data repurposing outlined in Chapter 2 lies the personal knowledge base that ties the other components of the potential system together. It is through this personal knowledge base that each e-scientist can describe their view on the data, and it is through this personal knowledge base that data driven disciplines can improve the quality of their publications.

Looking at the interactions between the personal knowledge base and the process of data repurposing immediately presents several directions of future work.

182 Automated combination Automatically combining selected data sources can be done through schema alignment and requires, among other things, robust deduplication. Great advances have been made in this direction with the increased popularity of big data and data science. The use of uncertainty for data integration approaches such as schema alignment has increased in recent research, and we are close to being able to produce an automatic first integration as put forward in Chapter 2 of this thesis.

Uncertainty as used in proposed solutions to data combination problems is mostly constrained to producing uncertainty. An investigation into the pervasive use of uncertainty throughout the combination process, accepting

uncertain inputs and reducing where possible the uncertainty in the output, seems a fruitful direction of work.

183 Integration explanation and exploration Increasing demands on reproducibility and justification in publications lead to greater demands on the e-scientist. Strong explanations from support tools combined with assisted exploration of data through exception finding and data profiling will allow the e-scientist to remain productive in the face of these increasing demands.

To support the e-scientist in their work any automated system should be able to justify and explain its output. Furthermore, these assistant systems should provide an intuitive interface tuned to the demands of the e-scientist. In-depth investigation of the current techniques and methods e-scientists use to explore integrated data can improve the way interactions with assistant systems are designed.

184 Feedback An automated system for improving the integration of multiple data sources requires feedback from the e-scientist. This feedback comes in the form of integration choices, but also in the form of rules and heuristics that indicate constraints and expectations.

Systems that can use previously found exceptions and patterns can solicit feedback from the e-scientist. How the e-scientist expresses these integration rules and expectations has a big influence on how well any potential system fits into their workflow. Both the interactions and methods used to express these rules, as well as their formal foundations, are fruitful directions for future work.

185 Sharing and collaboration In the process of doing research the e-scientist builds up a large personal knowledge base. Insights into the semantics and quality of used data sources is encoded in his integration rules and choices. These knowledge bases can be used as a basis for the reuse of effort through sharing and the creation of aggregate knowledge bases such as an ‘institute knowledge base’ to help new employees get started.

Investigating how integration rules from several knowledge bases can be combined, in what way this process may be automated, and what the impact of each personal knowledge base's subjective point of view is opens up the ability to easily share insights and reuse effort.

186 Expressing trust, opinion and conjecture Expressing the results of the integration as uncertain data opens up the option of representing integration conflicts as uncertainty. Instead of resolving the conflict the alternatives are all present in the intermediate results.

A good candidate for future work is the investigation of the ways uncertainty can be used to express other aspects of integrated data: representing trust, opinion and conjecture as uncertainty, and offering a formal model of how these concepts interact, will allow a more uniform handling of these concepts.

References

[1] Ziawasch Abedjan, Lukasz Golab and Felix Naumann. "Profiling relational data: a survey". In: *VLDB J.* 24.4 (2015), pp. 557–581. DOI: 10.1007/s00778-015-0389-y.

[2] Serge Abiteboul, Benny Kimelfeld, Yehoshua Sagiv and Pierre Senellart. "On the expressiveness of probabilistic XML models". In: *VLDB J.* 18.5 (2009), pp. 1041–1064. DOI: 10.1007/s00778-009-0146-1.

[3] J. K. Aggarwal and M. S. Ryoo. "Human activity analysis: A review". In: *ACM Comput. Surv.* 43.3 (2011), p. 16. DOI: 10.1145/1922649.1922653.

[4] Adrian M. Altenhoff and Christophe Dessimoz. "Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods". In: *PLoS Computational Biology* 5.1 (2009). DOI: 10.1371/journal.pcbi.1000262.

[5] Lyublena Antova, Christoph Koch and Dan Olteanu. "$10^{(10^6)}$ worlds and beyond: efficient representation and processing of incomplete information". In: *VLDB J.* 18.5 (2009), pp. 1021–1040. DOI: 10.1007/s00778-009-0149-y.

[6] Lyublena Antova, Thomas Jansen, Christoph Koch and Dan Olteanu. "Fast and Simple Relational Processing of Uncertain Data". In: *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*. Ed. by Gustavo Alonso, José A. Blakeley and Arbee L. P. Chen. IEEE, 2008, pp. 983–992. DOI: 10.1109/ICDE.2008.4497507.

[7] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak and Zachary G. Ives. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. Ed. by Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber and Philippe Cudré-Mauroux. Vol. 4825. Lecture Notes in Computer Science. Springer, 2007, pp. 722–735. ISBN: 978-3-540-76297-3. DOI: 10.1007/978-3-540-76298-0_52.

[8] Carlo Batini and Monica Scannapieco. *Data Quality: Concepts, Methodologies and Techniques*. Data-Centric Systems and Applications. Springer, 2006. ISBN: 978-3-540-33172-8.

[9] Niels Bloom, Mariët Theune and Franciska de Jong. "Document Categorization using Multilingual Associative Networks based on Wikipedia". In: *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*. Ed. by Aldo Gangemi, Stefano Leonardi and Alessandro Panconesi. ACM, 2015, pp. 841–846. ISBN: 978-1-4503-3473-0. DOI: 10.1145/2740908.2743003.

[10] Scott Boag, Michael Kay, Don Chamberlin, Jérôme Siméon, Mary F. Fernández, Jonathan Robie and Anders Berglund. *XML Path Language (XPath) 2.0 (Second Edition)*. W3C Recommendation. W3C, 2010. URL: http://www.w3.org/TR/2010/REC-xpath20-20101214/.

[11] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia and Anastasios Kementsietsidis. "Conditional Functional Dependencies for Data Cleaning". In: *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*. Ed. by Rada Chirkova, Asuman Dogac, M. Tamer Özsu and Timos K. Sellis. IEEE, 2007, pp. 746–755. DOI: 10.1109/ICDE.2007.367920.

[12] Rajendra Bose and James Frew. “Lineage retrieval for scientific data processing: a survey”. In: *ACM Comput. Surv.* 37.1 (2005), pp. 1–28. DOI: 10.1145/1057977.1057978.

[13] Randal E. Bryant. “Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams”. In: *ACM Comput. Surv.* 24.3 (1992), pp. 293–318. DOI: 10.1145/136035.136043.

[14] Michel Buffa, Fabien L. Gandon, Guillaume Erétéo, Peter Sander and Catherine Faron. “SweetWiki: A semantic wiki”. In: *J. Web Sem.* 6.1 (2008), pp. 84–97. DOI: 10.1016/j.websem.2007.11.003.

[15] Pablo Samuel Castro, Daqing Zhang, Chao Chen, Shijian Li and Gang Pan. “From taxi GPS traces to social and community dynamics: A survey”. In: *ACM Comput. Surv.* 46.2 (2013), p. 17. DOI: 10.1145/2543581.2543584.

[16] Stefano Ceri, Georg Gottlob and Letizia Tanca. *Logic Programming and Databases*. Springer, 1990. ISBN: 978-3-540-51728-3.

[17] Ugur Çetintemel, Stanley B. Zdonik, Donald Kossmann and Nesime Tatbul, eds. *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*. ACM, 2009. ISBN: 978-1-60558-551-2.

[18] You-Wei Cheah and Beth Plale. “Provenance Quality Assessment Methodology and Framework”. In: *J. Data and Information Quality* 5.3 (2015), 9:1–9:20. DOI: 10.1145/2665069.

[19] Weidong Chen, Terrance Swift and David Scott Warren. “Efficient Top-Down Computation of Queries under the Well-Founded Semantics”. In: *J. Log. Program.* 24.3 (1995), pp. 161–199. DOI: 10.1016/0743-1066(94)00028-5.

[20] *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*. www.cidrdb.org, 2007.

[21] Nino B. Cocchiarella and Max A. Freund. *Modal logic: an introduction to its syntax and semantics*. Oxford University Press, 2008. ISBN: 978-0-19-536657-0.

[22] Vítor Santos Costa, David Page and James Cussens. "CLP(*BN*): Constraint Logic Programming for Probabilistic Knowledge". In: *Probabilistic Inductive Logic Programming - Theory and Applications*. Ed. by Luc De Raedt, Paolo Frasconi, Kristian Kersting and Stephen Muggleton. Vol. 4911. Lecture Notes in Computer Science. Springer, 2008, pp. 156–188. ISBN: 978-3-540-78651-1. DOI: 10.1007/978-3-540-78652-8_6.

[23] Olivier Curé. "Improving the Data Quality of Drug Databases using Conditional Dependencies and Ontologies". In: *J. Data and Information Quality* 4.1 (2012), p. 3. DOI: 10.1145/2378016.2378019.

[24] Nilesh N. Dalvi, Christopher Ré and Dan Suciu. "Probabilistic databases: diamonds in the dirt". In: *Commun. ACM* 52.7 (2009), pp. 86–94. DOI: 10.1145/1538788.1538810.

[25] Daniel Davison. "A framework for creating semantic wikis for biomedical research laboratories". M.Sc. thesis. University of Twente, 2013. URL: http://eprints.eemcs.utwente.nl/24181/.

[26] Dimitar Denev, Arturas Mazeika, Marc Spaniol and Gerhard Weikum. "The SHARC framework for data quality in Web archiving". In: *VLDB J.* 20.2 (2011), pp. 183–207. DOI: 10.1007/s00778-011-0219-9.

[27] Steven DeRose and James Clark. *XML Path Language (XPath) Version 1.0*. W3C Recommendation. W3C, 1999. URL: http://www.w3.org/TR/1999/REC-xpath-19991116.

[28] *DokuWiki*. 2004–2016. URL: https://www.dokuwiki.org.

[29] Xin Luna Dong, Laure Berti-Equille and Divesh Srivastava. "Truth Discovery and Copying Detection in a Dynamic World". In: *PVLDB* 2.1 (2009), pp. 562–573. URL: http://www.vldb.org/pvldb/2/vldb09-335.pdf.

[30] Xin Luna Dong, Barna Saha and Divesh Srivastava. "Less is More: Selecting Sources Wisely for Integration". In: *PVLDB* 6.2 (2012), pp. 37–48. DOI: 10.14778/2535568.2448938. URL: http://www.vldb.org/pvldb/vol6/p37-dong.pdf.

[31] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun and Wei Zhang. "Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources". In: *PVLDB* 8.9 (2015), pp. 938–949. URL: http://www.vldb.org/pvldb/vol8/p938-dong.pdf.

[32] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis and Vassilios S. Verykios. "Duplicate Record Detection: A Survey". In: *IEEE Trans. Knowl. Data Eng.* 19.1 (2007), pp. 1–16. DOI: 10.1109/TKDE.2007.250581.

[33] Lujun Fang, Anish Das Sarma, Cong Yu and Philip Bohannon. "REX: Explaining Relationships between Entity Pairs". In: *PVLDB* 5.3 (2011), pp. 241–252. URL: http://www.vldb.org/pvldb/vol5/p241_lujunfang_vldb2012.pdf.

[34] Daniela Florescu, Jérôme Siméon, Scott Boag, Don Chamberlin, Mary F. Fernández and Jonathan Robie. *XQuery 1.0: An XML Query Language (Second Edition)*. W3C Recommendation. W3C, 2010. URL: http://www.w3.org/TR/2010/REC-xquery-20101214/.

[35] Python Software Foundation. *Python*. 1991–2016. URL: https://www.python.org.

[36] Ted Friedman and Michael Smith. *Measuring the Business Value of Data Quality*. Tech. rep. G00218962. Gartner, 2011.

[37] Norbert Fuhr. "Probabilistic datalog: Implementing logical information retrieval for advanced applications". In: *JASIS* 51.2 (2000), pp. 95–110. DOI: 10.1002/(SICI)1097-4571(2000)51:2<95::AID-ASI2>3.0.CO;2-H.

[38] *GeoNames*. 2006–2016. URL: http://geonames.org.

[39] François Goasdoué, Konstantinos Karanasos, Yannis Katsis, Julien Leblay, Ioana Manolescu and Stamatis Zampetakis. "Growing triples on trees: an XML-RDF hybrid model for annotated documents". In: *VLDB J.* 22.5 (2013), pp. 589–613. DOI: 10.1007/s00778-013-0321-2.

[40] Andreas Gohr. *Structured Data Plugin*. 2007–2016. URL: https://www.dokuwiki.org/plugin:data.

[41] Victor de Graaff, Dieter Pfoser, Maurice van Keulen and Rolf A. de By. "Spatiotemporal Behavior Profiling: A Treasure Hunt Case Study". In: *Web and Wireless Geographical Information Systems - 14th International Symposium, W2GIS 2015, Grenoble, France, May 21-22, 2015, Proceedings*. Ed. by Jérôme Gensel and Martin Tomko. Vol. 9080. Lecture Notes in Computer Science. Springer, 2015, pp. 143–158. ISBN: 978-3-319-18250-6. DOI: 10.1007/978-3-319-18251-3_9.

[42] Christian Grün, Alexander Holupirek and Marc H. Scholl. "Visually Exploring and Querying XML with BaseX". In: *Datenbanksysteme in Business, Technologie und Web (BTW 2007), 12. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), Proceedings, 7.-9. März 2007, Aachen, Germany*. Ed. by Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix and Christoph Brochhaus. Vol. 103. LNI. GI, 2007, pp. 629–632. ISBN: 978-3-88579-197-3. URL: http://subs.emis.de/LNI/Proceedings/Proceedings103/article1432.html.

[43] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang and Xingshe Zhou. "Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm". In: *ACM Comput. Surv.* 48.1 (2015), p. 7. DOI: 10.1145/2794400.

[44] Daniel Haas, Sanjay Krishnan, Jiannan Wang, Michael J. Franklin and Eugene Wu. "Wisteria: Nurturing Scalable Data Cleaning Infrastructure". In: *PVLDB* 8.12 (2015), pp. 2004–2015. URL: http://www.vldb.org/pvldb/vol8/p2004-haas.pdf.

[45] Mena B. Habib and Maurice van Keulen. "TwitterNEED: A hybrid approach for named entity extraction and disambiguation for tweet". In: *Natural Language Engineering* FirstView (July 2015), pp. 1–34. ISSN: 1469-8110. DOI: 10.1017/S1351324915000194.

[46] Mena B. Habib, Brend Wanders, Jan Flokstra and Maurice van Keulen. "Incremental Data Uncertainty Handling Using Evidence Combination: A Case Study on Maritime Data Reasoning". In: *26th International Workshop on Database and Expert Systems Applications, DEXA 2015, Valencia, Spain, September 1-4, 2015*. Ed. by Marcus Spies, Roland R. Wagner and A Min Tjoa. IEEE, 2015, pp. 147–151. ISBN: 978-1-4673-7581-8. DOI: 10.1109/DEXA.2015.45.

[47] Abbes Harati-Mokhtari, Alan Wall, Philip Brooks and Jin Wang. "Automatic Identification System (AIS): Data Reliability and Human Error Implications". In: *Journal of Navigation* 60 (03 Sept. 2007), pp. 373–389. ISSN: 1469-7785. DOI: 10.1017/S0373463307004298.

[48] International Conference on Harmonisation. *ICH E6: Good Clinical Practice*. ICH Guideline. Geneva, Switzerland, 1996.

[49] Dennis Heimann, Birgitta König-Ries and Jens Nieschulze. "The biodiversity-exploratory information system - Towards a service-oriented framework for knowledge-based data and tool integration". In: *Proc. of the Data Management Workshop*. 2009, pp. 65–73. DOI: 10.5880/TR32DB.KGA90.12.

[50] Melanie Herschel. "A Hybrid Approach to Answering Why-Not Questions on Relational Query Results". In: *J. Data and Information Quality* 5.3 (2015), 10:1–10:29. DOI: 10.1145/2665070.

[51] Tony Hey, Stewart Tansley and Kristin M. Tolle, eds. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009. ISBN: 978-0-9825442-0-4. URL: http://research.microsoft.com/en-us/collaboration/fourthparadigm/.

[52] Kyle S. Hickmann, Geoffrey Fairchild, Reid Priedhorsky, Nicholas Generous, James M. Hyman, Alina Deshpande and Sara Y. Del Valle. "Forecasting the 2013-2014 Influenza Season Using Wikipedia". In: *PLoS Computational Biology* 11.5 (2015). DOI: 10.1371/journal.pcbi.1004239.

[53] Victoria J. Hodge and Jim Austin. "A Survey of Outlier Detection Methodologies". In: *Artif. Intell. Rev.* 22.2 (2004), pp. 85–126. DOI: 10.1023/B:AIRE.0000045502.10941.a9.

[54] Paulien Hogeweg. "The Roots of Bioinformatics in Theoretical Biology". In: *PLoS Computational Biology* 7.3 (2011). Ed. by David B. Searls, e1002021. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.1002021.

[55] Emiel S. Hollander and Maurice van Keulen. "Storing and Querying Probabilistic XML Using a Probabilistic Relational DBMS". In: *Proceedings of the Fourth International VLDB workshop on Management of Uncertain Data (MUD 2010) in conjunction with VLDB 2010, Singapore, September 13, 2010.* Ed. by Ander de Keijzer and Maurice van Keulen. Vol. WP10-04. CTIT Workshop Proceedings Series. Centre for Telematics and Information Technology (CTIT), University of Twente, The Netherlands, 2010, pp. 35–49. URL: http://doc.utwente.nl/72454/.

[56] Jiewen Huang, Lyublena Antova, Christoph Koch and Dan Olteanu. "MayBMS: a probabilistic database management system". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009.* Ed. by Ugur Çetintemel, Stanley B. Zdonik, Donald Kossmann and Nesime Tatbul. ACM, 2009, pp. 1071–1074. ISBN: 978-1-60558-551-2. DOI: 10.1145/1559845.1559984.

[57] Stratos Idreos, Martin L. Kersten and Stefan Manegold. "Database Cracking". In: *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings.* www.cidrdb.org, 2007, pp. 68–78. URL: http://www.cidrdb.org/cidr2007/papers/cidr07p07.pdf.

[58] Ravi Jampani, Fei Xu, Mingxi Wu, Luis Leopoldo Perez, Christopher M. Jermaine and Peter J. Haas. "MCDB: A Monte Carlo Approach to Managing Uncertain Data". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. Ed. by Jason Tsong-Li Wang. ACM, 2008, pp. 687–700. ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376686.

[59] Guoqian Jiang, Harold R. Solbrig, Dave Iberson-Hurst, Rebecca D. Kush and Christopher G. Chute. "A collaborative framework for representation and harmonization of clinical study data elements using semantic MediaWiki". In: *Summit on Translational Bioinformatics* 2010 (2010), pp. 11–15. ISSN: 2153-6430.

[60] Maurice van Keulen. "Managing Uncertainty: The Road Towards Better Data Interoperability". In: *it - Information Technology* 54.3 (2012), pp. 138–146. DOI: 10.1524/itit.2012.0674.

[61] Maurice van Keulen and Mena B. Habib. "Handling Uncertainty in Information Extraction". In: *Proceedings of the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2011), Bonn, Germany, October 23, 2011*. Ed. by Fernando Bobillo, Rommel N. Carvalho, Paulo Cesar G. da Costa, Claudia d'Amato, Nicola Fanizzi, Kathryn B. Laskey, Thomas Lukasiewicz, Trevor Martin and Matthias Nickles. Vol. 778. CEUR Workshop Proceedings. CEUR-WS.org, 2011, pp. 109–112. URL: http://ceur-ws.org/Vol-778/pospaper3.pdf.

[62] Maurice van Keulen and Ander de Keijzer. "Qualitative effects of knowledge rules and user feedback in probabilistic data integration". In: *VLDB J.* 18.5 (2009), pp. 1191–1217. DOI: 10.1007/s00778-009-0156-z.

[63] Maurice van Keulen, Ander de Keijzer and Wouter Alink. "A Probabilistic XML Approach to Data Integration". In: *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*. Ed. by Karl Aberer, Michael J. Franklin

and Shojiro Nishio. IEEE Computer Society, 2005, pp. 459–470. ISBN: 978-0-7695-2285-2. DOI: 10.1109/ICDE.2005.11.

[64] Angelika Kimmig, Vítor Santos Costa, Ricardo Rocha, Bart Demoen and Luc De Raedt. "On the Efficient Execution of ProbLog Programs". In: *Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings*. Ed. by Maria Garcia de la Banda and Enrico Pontelli. Vol. 5366. Lecture Notes in Computer Science. Springer, 2008, pp. 175–189. ISBN: 978-3-540-89981-5. DOI: 10.1007/978-3-540-89982-2_22.

[65] Daniël Knippers. "Querying Uncertain Data in XML". M.Sc. thesis. University of Twente, 2014. URL: http://eprints.eemcs.utwente.nl/25897/.

[66] Christoph Koch and Dan Olteanu. "Conditioning probabilistic databases". In: *PVLDB* 1.1 (2008), pp. 313–325. URL: http://www.vldb.org/pvldb/1/1453894.pdf.

[67] Eugene V. Koonin. "Orthologs, paralogs, and evolutionary genomics". In: *Annual Review of Genetics* 39 (2005), pp. 309–338.

[68] Markus Krötzsch, Denny Vrandecic and Max Völkel. "Semantic MediaWiki". In: *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*. Ed. by Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold and Lora Aroyo. Vol. 4273. Lecture Notes in Computer Science. Springer, 2006, pp. 935–942. ISBN: 978-3-540-49029-6. DOI: 10.1007/11926078_68.

[69] Jasper Kuperus, Cor J. Veenman and Maurice van Keulen. "Increasing NER Recall with Minimal Precision Loss". In: *2013 European Intelligence and Security Informatics Conference, Uppsala, Sweden, August 12-14, 2013*. IEEE, 2013, pp. 106–111. ISBN: 978-0-7695-5062-6. DOI: 10.1109/EISIC.2013.23.

[70] Lukasz A. Kurgan and Petr Musílek. "A survey of Knowledge Discovery and Data Mining process models". In: *Knowledge Eng. Review* 21.1 (2006), pp. 1–24. DOI: 10.1017/S0269888906000737.

[71] Arnold Kuzniar, Ke Lin, Ying He, Harm Nijveen, Sándor Pongor and Jack A.M. Leunissen. "ProGMap: an integrated annotation resource for protein orthology". In: *Nucleic Acids Research* 37.suppl 2 (2009), W428–W434. DOI: 10.1093/nar/gkp462.

[72] Arnold Kuzniar, Roeland C.H.J. van Ham, Sándor Pongor and Jack A.M. Leunisse. "The quest for orthologs: finding the corresponding gene across genomes". In: *Trends in Genetics* 24.11 (2008), pp. 539 –551. ISSN: 0168-9525. DOI: 10.1016/j.tig.2008.08.009.

[73] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer and Christian Bizer. "DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia". In: *Semantic Web* 6.2 (2015), pp. 167–195. DOI: 10.3233/SW-140134.

[74] Lawrence Lessig. *Code 2.0*. 2nd. 387 Park Avenue South, New York NY 10016-8810: Basic Books, 2006. ISBN: 978-0-465-03914-2. URL: http://codev2.cc/download+remix/Lessig-Codev2.pdf.

[75] Matteo Magnani and Danilo Montesi. "A Survey on Uncertainty Management in Data Integration". In: *J. Data and Information Quality* 2.1 (2010). DOI: 10.1145/1805286.1805291.

[76] Gonzalo Mariscal, Óscar Marbán and Covadonga Fernández. "A survey of data mining and knowledge discovery process models and methodologies". In: *Knowledge Eng. Review* 25.2 (2010), pp. 137–166. DOI: 10.1017/S0269888910000032.

[77] Timm Meiser, Maximilian Dylla and Martin Theobald. "Interactive reasoning in uncertain RDF knowledge bases". In: *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*. Ed. by

Craig Macdonald, Iadh Ounis and Ian Ruthven. ACM, 2011, pp. 2557–2560. ISBN: 978-1-4503-0717-8. DOI: 10.1145/2063576.2064018.

[78] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F. Moore and Dan Suciu. "The Complexity of Causality and Responsibility for Query Answers and non-Answers". In: *PVLDB* 4.1 (2010), pp. 34–45. URL: http://www.vldb.org/pvldb/vol4/p34-meliou.pdf.

[79] Eirinaios Michelakis, Rajasekar Krishnamurthy, Peter J. Haas and Shivakumar Vaithyanathan. "Uncertainty management in rule-based information extraction systems". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*. Ed. by Ugur Çetintemel, Stanley B. Zdonik, Donald Kossmann and Nesime Tatbul. ACM, 2009, pp. 101–114. ISBN: 978-1-60558-551-2. DOI: 10.1145/1559845.1559858.

[80] Davide Mottin, Matteo Lissandrini, Yannis Velegrakis and Themis Palpanas. "Exemplar Queries: Give me an Example of What You Need". In: *PVLDB* 7.5 (2014), pp. 365–376. URL: http://www.vldb.org/pvldb/vol7/p365-mottin.pdf.

[81] Stephen Muggleton. "Stochastic logic programs". In: *Advances in inductive logic programming* 32 (1996). Ed. by Luc de Raedt, pp. 254–264.

[82] Heiko Müller, Johann Christoph Freytag and Ulf Leser. "Improving data quality by source analysis". In: *J. Data and Information Quality* 2.4 (2012), p. 15. DOI: 10.1145/2107536.2107538.

[83] Michi Mutsuzaki, Martin Theobald, Ander de Keijzer, Jennifer Widom, Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Raghotham Murthy and Tomoe Sugihara. "Trio-One: Layering Uncertainty and Lineage on a Conventional DBMS (Demo)". In: *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*. www.cidrdb.org, 2007, pp. 269–274. URL: http://www.cidrdb.org/cidr2007/papers/cidr07p30.pdf.

[84] NCBI Resource Coordinators. “Database resources of the National Center for Biotechnology Information”. In: *Nucleic Acids Research* 41.D1 (2013), pp. D8–D20. DOI: 10.1093/nar/gks1189.

[85] Dan Olteanu, Jiewen Huang and Christoph Koch. “Approximate confidence computation in probabilistic databases”. In: *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*. Ed. by Feifei Li, Mirella M. Moro, Shahram Ghandeharizadeh, Jayant R. Haritsa, Gerhard Weikum, Michael J. Carey, Fabio Casati, Edward Y. Chang, Ioana Manolescu, Sharad Mehrotra, Umeshwar Dayal and Vassilis J. Tsotras. IEEE, 2010, pp. 145–156. ISBN: 978-1-4244-5444-0. DOI: 10.1109/ICDE.2010.5447826.

[86] Dan Olteanu, Jiewen Huang and Christoph Koch. “SPROUT: Lazy vs. Eager Query Plans for Tuple-Independent Probabilistic Databases”. In: *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*. Ed. by Yannis E. Ioannidis, Dik Lun Lee and Raymond T. Ng. IEEE, 2009, pp. 640–651. ISBN: 978-0-7695-3545-6. DOI: 10.1109/ICDE.2009.123.

[87] Fabian Panse. “Duplicate Detection in Probabilistic Relational Databases”. eng. PhD thesis. Von-Melle-Park 3, 20146 Hamburg: University of Hamburg, 2014. URL: http://ediss.sub.uni-hamburg.de/volltexte/2015/7430.

[88] Fabian Panse, Maurice van Keulen and Norbert Ritter. “Indeterministic Handling of Uncertain Decisions in Deduplication”. In: *J. Data and Information Quality* 4.2 (2013), p. 9. DOI: 10.1145/2435221.2435225.

[89] Class-Thido Pfaff, Birgitta König-Ries, Anne C. Lang, Sophia Ratcliffe, Christian Wirth, Xingxing Man and Karin Nadrowski. “rBEFdata: documenting data exchange and analysis for a collaborative data management platform”. In: *Ecology and Evolution* 5.14 (2015), pp. 2890–2897. ISSN: 2045-7758. DOI: 10.1002/ece3.1547.

[90] David Poole. “Probabilistic Horn Abduction and Bayesian Networks”. In: *Artificial Intelligence* 64.1 (1993), pp. 81–129. DOI: 10.1016/0004-3702(93)90061-F.

[91] Sean Powell, Damian Szklarczyk, Kalliopi Trachana, Alexander Roth, Michael Kuhn, Jean Muller, Ronald Arnold, Thomas Rattei, Ivica Letunic, Tobias Doerks, Lars J. Jensen, Christian von Mering and Peer Bork. “eggNOG v3. 0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges”. In: *Nucleic Acids Research* 40 (2011). DOI: 10.1093/nar/gkr1060.

[92] Luc De Raedt, Angelika Kimmig and Hannu Toivonen. “ProbLog: A Probabilistic Prolog and Its Application in Link Discovery”. In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. Ed. by Manuela M. Veloso. 2007, pp. 2462–2467. URL: http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-396.pdf.

[93] Matthew Richardson and Pedro M. Domingos. “Markov logic networks”. In: *Machine Learning* 62.1-2 (2006), pp. 107–136. DOI: 10.1007/s10994-006-5833-1.

[94] Tjitze D. Rienstra. “Dealing with uncertainty in the semantic web”. M.Sc. thesis. University of Twente, 2009. URL: http://eprints.eemcs.utwente.nl/16480/.

[95] Loïc Royer, Matthias Reimann, Bill Andreopoulos and Michael Schroeder. “Unraveling Protein Networks with Power Graph Analysis”. In: *PLoS Computational Biology* 4.7 (2008). DOI: 10.1371/journal.pcbi.1000108.

[96] Bertrand Russell. “Vagueness”. In: *Australasian Journal of Philosophy* 1.2 (1923), pp. 84–92.

[97] Taisuke Sato and Yoshitaka Kameya. “Parameter Learning of Logic Programs for Symbolic-Statistical Modeling”. In: *Journal of Artificial*

Intelligence Research (JAIR) 15 (2001), pp. 391–454. DOI: 10.1613/jair.912.

[98] Sebastian Schaffert. "IkeWiki: A Semantic Wiki for Collaborative Knowledge Management". In: *15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE) 2006, 26-28 June 2006, Manchester, United Kingdom*. IEEE Computer Society, 2006, pp. 388–396. ISBN: 978-0-7695-2623-2. DOI: 10.1109/WETICE.2006.46.

[99] Leonard J. Seligman, Arnon Rosenthal, Paul E. Lehner and Angela Smith. "Data Integration: Where Does the Time Go?" In: *IEEE Data Eng. Bull.* 25.3 (2002), pp. 3–10. URL: http://sites.computer.org/debull/A02SEP-CD.pdf.

[100] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976. ISBN: 978-0-691-10042-5.

[101] C. Shearer. "The CRISP-DM model: The new blueprint for data mining". In: *J. of Data Warehousing* 5.4 (2000), pp. 13–22.

[102] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang and Christopher Ré. "Incremental Knowledge Base Construction Using DeepDive". In: *PVLDB* 8.11 (2015), pp. 1310–1321. URL: http://www.vldb.org/pvldb/vol8/p1310-shin.pdf.

[103] John Snelson, Don Chamberlin, Michael Dyck and Jonathan Robie. *XML Path Language (XPath) 3.0*. W3C Recommendation. W3C, 2014. URL: http://www.w3.org/TR/2014/REC-xpath-30-20140408/.

[104] Shaoxu Song, Aoqian Zhang, Lei Chen and Jianmin Wang. "Enriching Data Imputation with Extensive Similarity Neighbors". In: *PVLDB* 8.11 (2015), pp. 1286–1297. URL: http://www.vldb.org/pvldb/vol8/p1286-song.pdf.

[105] Sjoerd van der Spoel, Maurice van Keulen and Chintan Amrit. "Process Prediction in Noisy Data Sets: A Case Study in a Dutch Hospital". In: *Data-Driven Process Discovery and Analysis - Second IFIP WG 2.6, 2.12 International Symposium, SIMPDA 2012, Campione d'Italia, Italy, June 18-20, 2012, Revised Selected Papers*. Ed. by Philippe Cudré-Mauroux, Paolo Ceravolo and Dragan Gasevic. Vol. 162. Lecture Notes in Business Information Processing. Springer, 2012, pp. 60–83. ISBN: 978-3-642-40918-9. DOI: 10.1007/978-3-642-40919-6_4.

[106] International Organization for Standardization. *ISO 8601:2004. Data Elements and Interchange Formats – Information Exchange – Representation of Dates and Times*. ISO Standard. Geneva, Switzerland, 2004.

[107] Paul Stapersma. "Efficient Query Evaluation on Probabilistic XML Data". M.Sc. thesis. University of Twente, 2012. URL: http://eprints.eemcs.utwente.nl/22658/.

[108] Miranda D. Stobbe, Sander M. Houten, Antoine H. C. van Kampen, Ronald J. A. Wanders and Perry D. Moerland. "Improving the description of metabolic networks: the TCA cycle as example". In: *The FASEB Journal* 26.9 (2012), pp. 3625–3636. DOI: 10.1096/fj.11-203091.

[109] Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum. "Yago: a core of semantic knowledge". In: *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. Ed. by Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider and Prashant J. Shenoy. ACM, 2007, pp. 697–706. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242667.

[110] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao and Chung Wu. "Recovering Semantics of Tables on the Web". In: *PVLDB* 4.9 (2011), pp. 528–538. URL: http://www.vldb.org/pvldb/vol4/p528-venetis.pdf.

[111] *VesselFinder*. 2011–2016. URL: http://www.vesselfinder.com/.

[112] Denny Vrandecic. "Wikidata: a new platform for collaborative data collection". In: *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*. Ed. by Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich and Steffen Staab. ACM, 2012, pp. 1063–1064. ISBN: 978-1-4503-1230-1. DOI: 10.1145/2187980.2188242.

[113] Brend Wanders and Steven te Brinke. "Strata: Typed Semi-Structured Data in DokuWiki". In: *Proceedings of The International Symposium on Open Collaboration, OpenSym 2014, Berlin, Germany, August 27 - 29, 2014*. Ed. by Dirk Riehle, Jesús M. González-Barahona, Gregorio Robles, Kathrin M. Möslein, Ina Schieferdecker, Ulrike Cress, Astrid Wichmann, Brent Hecht and Nicolas Jullien. ACM, 2014, 50:1–50:2. ISBN: 978-1-4503-3016-9. DOI: 10.1145/2641580.2641636.

[114] Brend Wanders and Maurice van Keulen. "Revisiting the formal foundation of Probabilistic Databases". In: *2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15), Gijón, Spain., June 30, 2015*. Ed. by José M. Alonso, Humberto Bustince and Marek Reformat. Atlantis Press, 2015. ISBN: 978-94-62520-77-6. DOI: 10.2991/ifsa-eusflat-15.2015.43.

[115] Brend Wanders, Maurice van Keulen and Jan Flokstra. "JudgeD: a Probabilistic Datalog with Dependencies". In: *Proceedings of the Workshop on Declarative Learning Based Programming, DeLBP 2016, Phoenix, AZ, USA*. Phoenix, AZ, USA: AAAI Press, 2016. URL: http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12564.

[116] Brend Wanders, Maurice van Keulen and Paul E. van der Vet. "Uncertain Groupings: Probabilistic Combination of Grouping Data". In: *Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part I*. Ed. by Qiming Chen, Abdelkader Hameurlain, Farouk Toumani, Roland Wagner and Hendrik Decker. Vol. 9261. Lecture Notes in Computer

Science. Springer, 2015, pp. 236–250. ISBN: 978-3-319-22848-8. DOI: 10.1007/978-3-319-22849-5_17.

[117] Steven Euijong Whang and Hector Garcia-Molina. "Incremental entity resolution on rules and data". In: *VLDB J.* 23.1 (2014), pp. 77–102. DOI: 10.1007/s00778-013-0315-0.

[118] *Wikipedia*. 2001–2016. URL: https://www.wikipedia.org.

[119] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a standard process model for data mining". In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. 2000, pp. 29–39.

[120] Cathy H. Wu, Anastasia Nikolskaya, Hongzhan Huang, Lai-Su L. Yeh, Darren A. Natale, C. R. Vinayaka, Zhang-Zhi Hu, Raja Mazumder, Sandeep Kumar, Panagiotis Kourtesis, Robert S. Ledley, Baris E. Suzek, Leslie Arminski, Yongxing Chen, Jian Zhang, Jorge Louie Cardenas, Sehee Chung, Jorge Castro-Alvear, Georgi Dinkov and Winona C. Barker. "PIRSF: family classification system at the Protein Information Resource". In: *Nucleic Acids Research* 32.suppl 1 (2004), pp. D112–D114. DOI: 10.1093/nar/gkh097.

[121] Adnan Yazici and Roy George. *Fuzzy database modeling*. Vol. 26. Studies in Fuzziness and Soft Computing. Physica-Verlag Heidelberg, 1999. ISBN: 978-3-7908-1171-1. DOI: 10.1007/978-3-7908-1880-2.

[122] Barry Zeeberg, Joseph Riss, David W. Kane, Kimberly J. Bussey, Edward Uchio, W. Marston Linehan, J. Carl Barrett and John N. Weinstein. "Mistaken Identifiers: Gene name errors can be introduced inadvertently when using Excel in bioinformatics". In: *BMC Bioinformatics* 5 (2004), p. 80. DOI: 10.1186/1471-2105-5-80.

Summary / Samenvatting

Besides the scientific paradigms of empiricism, mathematical modelling, and simulation, the method of combining and analysing data in novel ways has become a main research paradigm capable of tackling research questions that could not be answered before. To speed up research in this new paradigm, scientists are reusing and integrating data originally gathered for different purposes. This repurposing of data requires a thorough understanding of the used data sources. Data understanding is an ongoing process in which the scientists gains insight into the semantics and quality of the data through exploration and use.

In this book we propose a flexible method to guide this exploration and to highlight the places where automated assistance can be used to the greatest effect. The method is based on the principles of 'good is good enough' and 'pay as you go', meaning that the scientist puts in only as much effort as is necessary to get the integrated data to the level of quality that he needs to continue his research. This book pursues two directions of research.

The first is an investigation of note taking. By documenting his exploration efforts the scientist can share his understanding of the data sources with others. To support the scientist in this a prototype note taking system is created. This system offers a compromise between the exploratory workflow of the scientist and the rigid procedures of the research institute.

The second direction is the use of probabilistic data to support the 'pay as you go' principle. A formal framework for the creation of probabilistic data models is introduced. By keeping data accessible even if there are contradictions or multiple alternatives, the scientists can postpone data integration choices that would have otherwise prevented him from continuing with his work.

Samenvatting

Naast de wetenschappelijke paradigma's van empirisme, wiskundige modellering en simulatie is het op nieuwe manieren combineren en analyseren van gegevens een belangrijke onderzoeksmethode geworden waarmee onbeantwoorde onderzoeksvragen opgelost kunnen worden. Om onderzoek volgens deze nieuwe methode te versnellen maken wetenschappers vaak gebruik van gegevens die voor hele andere doeleinden zijn verzameld. Het begrijpen van deze gegevens is een doorlopend proces waarmee de wetenschapper inzicht verkrijgt in de semantiek en kwaliteit van bronnen door deze te verkennen en te gebruiken.

In dit boek stellen we een flexibele methode voor om verkenning en gebruik te begeleiden en plekken aan te wijzen waar automatische ondersteuning het meest effectief kan zijn. Deze methode is gebaseerd op de principes van 'goed is goed genoeg' en 'pas betalen bij gebruik', waarbij de wetenschapper slechts dat doet wat noodzakelijk is om de kwaliteit van de informatie te verbeteren tot hij verder kan gaan met zijn onderzoek. Dit boek volgt twee onderzoeksrichtingen.

De eerste is het onderzoeken van het proces waarmee notities gemaakt worden. Door het pad waarmee de wetenschapper gegevens verkent te documenteren kan hij zijn begrip van de informatiebron delen met anderen. Om de wetenschapper hierbij te ondersteunen is een prototype gemaakt van een systeem waarmee dergelijke notities bijgehouden kunnen worden. Dit systeem bied een compromis tussen de verkennende werkwijze van de wetenschapper en de rigide procedures van zijn onderzoeksinstituut.

De tweede onderzoeksrichting is het gebruik van probabilistische informatie voor het ondersteunen van het 'pas betalen bij gebruik' principe. Er word een formeel kader voor het maken van probabilistische datamodellen voorgesteld. Door informatie toegankelijk te houden, zelfs als er verschillen of tegenstrijdigheden zijn, kan de wetenschapper het oplossen van dergelijke conflicten uitstellen, terwijl dit hem anders in de voortgang had belemmerd.

SIKS Dissertation Series

1998

1 Johan van den Akker (CWI) *DEGAS: An Active, Temporal Database of Autonomous Objects*

2 Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*

3 Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of Business Conversations*

4 Dennis Breuker (UM) *Memory versus Search in Games*

5 E. W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

1 Mark Sloof (VUA) *Physiology of Quality Change Modelling: Automated modelling of*

2 Rob Potharst (EUR) *Classification using decision trees and neural nets*

3 Don Beal (UM) *The Nature of Minimax Search*

4 Jacques Penders (UM) *The practical Art of Moving Physical Objects*

5 Aldo de Moor (KUB) *Empowering Communities: A Method for the Legitimate User-Driven*

6 Niek J. E. Wijngaards (VUA) *Re-design of compositional systems*

7 David Spelt (UT) *Verification support for object database design*

8 Jacques H. J. Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism*

2000

1 Frank Niessink (VUA) *Perspectives on Improving Software Maintenance*

2 Koen Holtman (TUe) *Prototyping of CMS Storage Management*

3 Carolien M. T. Metselaar (UvA) *Sociaal-organisatorische gevolgen van kennistechnologie*

4 Geert de Haan (VUA) *ETAG, A Formal Model of Competence Knowledge for User Interface*

5 Ruud van der Pol (UM) *Knowledge-based Query Formulation in Information Retrieval*

6 Rogier van Eijk (UU) *Programming Languages for Agent Communication*

7 Niels Peek (UU) *Decision-theoretic Planning of Clinical Patient Management*

8 Veerle Coupé (EUR) *Sensitivity Analyis of Decision-Theoretic Networks*

9 Florian Waas (CWI) *Principles of Probabilistic Query Optimization*

10 Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*

11 Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

1 Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*

2 Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*

3 Maarten van Someren (UvA) *Learning as problem solving*

4 Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

5 Jacco van Ossenbruggen (VUA) *Processing Structured Hypermedia: A Matter of Style*

6 Martijn van Welie (VUA) *Task-based User Interface Design*

7 Bastiaan Schonhage (VUA) *Diva: Architectural Perspectives on Information Visualization*

8 Pascal van Eck (VUA) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*

9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models*

10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice*

11 Tom M. van Engers (VUA) *Knowledge Management*

2002

1 Nico Lassing (VUA) *Architecture-Level Modifiability Analysis*

2 Roelof van Zwol (UT) *Modelling and searching web-based document collections*

3 Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*

4 Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*

5 Radu Serban (VUA) *The Private Cyberspace Modeling Electronic*

6 Laurens Mommers (UL) *Applied legal epistemology: Building a knowledge-based ontology of*

7 Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive*

8 Jaap Gordijn (VUA) *Value Based Requirements Engineering: Exploring Innovative*

9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy*

10 Brian Sheppard (UM) *Towards Perfect Play of Scrabble*

11 Wouter C. A. Wijngaards (VUA) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

12 Albrecht Schmidt (UvA) *Processing XML in Database Systems*

13 Hongjing Wu (TUe) *A Reference Architecture for Adaptive Hypermedia Applications*

14 Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

15 Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

16 Pieter van Langen (VUA) *The Anatomy of Design: Foundations, Models and Applications*

17 Stefan Manegold (UvA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

1 Heiner Stuckenschmidt (VUA) *Ontology-Based Information Sharing in Weakly Structured Environments*

2 Jan Broersen (VUA) *Modal Action Logics for Reasoning About Reactive Systems*

3 Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

4 Milan Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*

5 Jos Lehmann (UvA) *Causation in Artificial Intelligence and Law: A modelling approach*

6 Boris van Schooten (UT) *Development and specification of virtual environments*

7 Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*

8 Yongping Ran (UM) *Repair Based Scheduling*

9 Rens Kortmann (UM) *The resolution of visually guided behaviour*

10 Andreas Lincke (UvT) *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

11 Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

12 Roeland Ordelman (UT) *Dutch speech recognition in multimedia information retrieval*

13 Jeroen Donkers (UM) *Nosce Hostem: Searching with Opponent Models*

14 Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

15 Mathijs de Weerdt (TUD) *Plan Merging in Multi-Agent Systems*

16 Menzo Windhouwer (CWI) *Feature Grammar Systems: Incremental Maintenance of Indexes to Digital Media Warehouses*

17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

18 Levente Kocsis (UM) *Learning Search Decisions*

2004

1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*

3 Perry Groot (VUA) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

4 Chris van Aart (UvA) *Organizational Principles for Multi-Agent Architectures*

5 Viara Popova (EUR) *Knowledge discovery and monotonicity*

6 Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*

7 Elise Boltjes (UM) *Voorbeeldig onderwijs: voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

8 Joop Verbeek (UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*

9 Martin Caminada (VUA) *For the Sake of the Argument: explorations into argument-based reasoning*

10 Suzanne Kabel (UvA) *Knowledge-rich indexing of learning-objects*

11 Michel Klein (VUA) *Change Management for Distributed Ontologies*

12 The Duy Bui (UT) *Creating emotions and facial expressions for embodied agents*

13 Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*

14 Paul Harrenstein (UU) *Logic in Conflict. Logical Explorations in Strategic Equilibrium*

15 Arno Knobbe (UU) *Multi-Relational Data Mining*

16 Federico Divina (VUA) *Hybrid Genetic Relational Search for Inductive Learning*

17 Mark Winands (UM) *Informed Search in Complex Games*

18 Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*

19 Thijs Westerveld (UT) *Using generative probabilistic models for multimedia retrieval*

20 Madelon Evers (Nyenrode) *Learning from Design: facilitating multidisciplinary design teams*

2005

1 Floor Verdenius (UvA) *Methodological Aspects of Designing Induction-Based Applications*

2 Erik van der Werf (UM) *AI techniques for the game of Go*

3 Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*

4 Nirvana Meratnia (UT) *Towards Database Support for Moving Object data*

5 Gabriel Infante-Lopez (UvA) *Two-Level Probabilistic Grammars for Natural Language Parsing*

6 Pieter Spronck (UM) *Adaptive Game AI*

7 Flavius Frasincar (TUe) *Hypermedia Presentation Generation for Semantic Web Information Systems*

8 Richard Vdovjak (TUe) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*

9 Jeen Broekstra (VUA) *Storage, Querying and Inferencing for Semantic Web Languages*

10 Anders Bouwer (UvA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*

11 Elth Ogston (VUA) *Agent Based Matchmaking and Clustering: A Decentralized Approach to Search*

12 Csaba Boer (EUR) *Distributed Simulation in Industry*

13 Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*

14 Borys Omelayenko (VUA) *Web-Service configuration on the Semantic Web: Exploring how semantics meets pragmatics*

15 Tibor Bosse (VUA) *Analysis of the Dynamics of Cognitive Processes*

16 Joris Graaumans (UU) *Usability of XML Query Languages*

17 Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*

18 Danielle Sent (UU) *Test-selection strategies for probabilistic networks*

19 Michel van Dartel (UM) *Situated Representation*

20 Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*

21 Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

1 Samuil Angelov (TUe) *Foundations of B2B Electronic Contracting*

2 Cristina Chisalita (VUA) *Contextual issues in the design and use of information technology in organizations*

3 Noor Christoph (UvA) *The role of metacognitive skills in learning to solve problems*

4 Marta Sabou (VUA) *Building Web Service Ontologies*

5 Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*

6 Ziv Baida (VUA) *Software-aided Service Bundling: Intelligent Methods & Tools for Graphical Service Modeling*

7 Marko Smiljanic (UT) *XML schema matching: balancing efficiency and effectiveness by means of clustering*

8 Eelco Herder (UT) *Forward, Back and Home Again: Analyzing User Behavior on the Web*

9 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*

10 Ronny Siebes (VUA) *Semantic Routing in Peer-to-Peer Systems*

11 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*

12 Bert Bongers (VUA) *Interactivation: Towards an e-cology of people, our technological environment, and the arts*

13 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*

14 Johan Hoorn (VUA) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*

15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*

16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*

17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*

18 Valentin Zhizhkun (UvA) *Graph transformation for Natural Language Processing*

19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*

20 Marina Velikova (UvT) *Monotone models for prediction in data mining*

21 Bas van Gils (RUN) *Aptness on the Web*

22 Paul de Vrieze (RUN) *Fundaments of Adaptive Personalisation*

23 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*

24 Laura Hollink (VUA) *Semantic Annotation for Retrieval of Visual Resources*

25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*

26 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*

27 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*

28 Borkur Sigurbjornsson (UvA) *Focused Information Access using XML Element Retrieval*

2007

1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures*

2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*

3 Peter Mika (VUA) *Social Networks and the Semantic Web*

4 Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*

5 Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*

6 Gilad Mishne (UvA) *Applied Text Analytics for Blogs*

7 Natasa Jovanovic' (UT) *To Whom It May Concern: Addressee Identification in Face-to-Face Meetings*

8 Mark Hoogendoorn (VUA) *Modeling of Change in Multi-Agent Organizations*

9 David Mobach (VUA) *Agent-Based Mediated Service Negotiation*

10 Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*

11 Natalia Stash (TUe) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*

12 Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*

13 Rutger Rienks (UT) *Meetings in Smart Environments: Implications of Progressing Technology*

14 Niek Bergboer (UM) *Context-Based Image Analysis*

15 Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*

16 Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*

17 Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*

18 Bart Orriens (UvT) *On the development an management of adaptive business collaborations*

19 David Levy (UM) *Intimate relationships with artificial partners*

20 Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*

21 Karianne Vermaas (UU) *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*

22 Zlatko Zlatev (UT) *Goal-oriented design of value and process models from patterns*

23 Peter Barna (TUe) *Specification of Application Logic in Web Information Systems*

24 Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*

25 Joost Schalken (VUA) *Empirical Investigations in Software Process Improvement*

2008

1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*

2 Alexei Sharpanskykh (VUA) *On Computer-Aided Methods for Modeling and Analysis of Organizations*

3 Vera Hollink (UvA) *Optimizing hierarchical menus: a usage-based approach*

4 Ander de Keijzer (UT) *Management of Uncertain Data: towards unattended integration*

5 Bela Mutschler (UT) *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*

6 Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*

7 Peter van Rosmalen (OU) *Supporting the tutor in the design and support of adaptive e-learning*

8 Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*

9 Christof van Nimwegen (UU) *The paradox of the guided user: assistance can be counter-effective*

10 Wauter Bosma (UT) *Discourse oriented summarization*

11 Vera Kartseva (VUA) *Designing Controls for Network Organizations: A Value-Based Approach*

12 Jozsef Farkas (RUN) *A Semiotically Oriented Cognitive Model of Knowledge Representation*

13 Caterina Carraciolo (UvA) *Topic Driven Access to Scientific Handbooks*

14 Arthur van Bunningen (UT) *Context-Aware Querying: Better Answers with Less Effort*

15 Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains*

16 Henriette van Vugt (VUA) *Embodied agents from a user's perspective*

17 Martin Op 't Land (TUD) *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*

18 Guido de Croon (UM) *Adaptive Active Vision*

19 Henning Rode (UT) *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*

20 Rex Arendsen (UvA) *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven*

21 Krisztian Balog (UvA) *People Search in the Enterprise*

22 Henk Koning (UU) *Communication of IT-Architecture*

23 Stefan Visscher (UU) *Bayesian network models for the management of ventilator-associated pneumonia*

24 Zharko Aleksovski (VUA) *Using background knowledge in ontology matching*

25 Geert Jonker (UU) *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*

26 Marijn Huijbregts (UT) *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*

27 Hubert Vogten (OU) *Design and Implementation Strategies for IMS Learning Design*

28 Ildiko Flesch (RUN) *On the Use of Independence Relations in Bayesian Networks*

29 Dennis Reidsma (UT) *Annotations and Subjective Machines: Of Annotators, Embodied Agents, Users, and Other Humans*

30 Wouter van Atteveldt (VUA) *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*

31 Loes Braun (UM) *Pro-Active Medical Information Retrieval*

32 Trung H. Bui (UT) *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*

33 Frank Terpstra (UvA) *Scientific Workflow Design: theoretical and practical issues*

34 Jeroen de Knijf (UU) *Studies in Frequent Tree Mining*

35 Ben Torben Nielsen (UvT) *Dendritic morphologies: function shapes structure*

2009

1 Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*

2 Willem Robert van Hage (VUA) *Evaluating Ontology-Alignment Techniques*

3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*

4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks: Based on Knowledge, Cognition, and Quality*

6 Muhammad Subianto (UU) *Understanding Classification*

7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (VUA) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*

10 Jan Wielemaker (UvA) *Logic programming for knowledge-intensive interactive applications*

11 Alexander Boer (UvA) *Legal Theory, Sources of Law & the Semantic Web*

12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin) *Operating Guidelines for Services*

13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*

14 Maksym Korotkiy (VUA) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*

15 Rinke Hoekstra (UvA) *Ontology Representation: Design Patterns and Ontologies that Make Sense*

16 Fritz Reul (UvT) *New Architectures in Computer Chess*

17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*

18 Fabian Groffen (CWI) *Armada, An Evolving Database System*

19 Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

20 Bob van der Vecht (UU) *Adjustable Autonomy: Controling Influences on Decision Making*

21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*

22 Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*

23 Peter Hofgesang (VUA) *Modelling Web Usage in a Changing Environment*

24 Annerieke Heuvelink (VUA) *Cognitive Models for Training Simulations*

25 Alex van Ballegooij (CWI) *RAM: Array Database Management through Relational Mapping*

26 Fernando Koch (UU) *An Agent-Based Model for the Development of Intelligent Mobile Services*

27 Christian Glahn (OU) *Contextual Support of social Engagement and Reflection on the Web*

28 Sander Evers (UT) *Sensor Data Management with Probabilistic Models*

29 Stanislav Pokraev (UT) *Model-Driven Semantic Integration of Service-Oriented Applications*

30 Marcin Zukowski (CWI) *Balancing vectorized query execution with bandwidth-optimized storage*

31 Sofiya Katrenko (UvA) *A Closer Look at Learning Relations from Text*

32 Rik Farenhorst (VUA) *Architectural Knowledge Management: Supporting Architects and Auditors*

33 Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*

34 Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*

35 Wouter Koelewijn (UL) *Privacy en Politiegegevens: Over geautomatiseerde normatieve informatie-uitwisseling*

36 Marco Kalz (OUN) *Placement Support for Learners in Learning Networks*

37 Hendrik Drachsler (OUN) *Navigation Support for Learners in Informal Learning Networks*

38 Riina Vuorikari (OU) *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*

39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) *Service Substitution: A Behavioral Approach Based on Petri Nets*

40 Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*

41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*

42 Toine Bogers (UvT) *Recommender Systems for Social Bookmarking*

43 Virginia Nunes Leal Franqueira (UT) *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*

44 Roberto Santana Tapia (UT) *Assessing Business-IT Alignment in Networked Organizations*

45 Jilles Vreeken (UU) *Making Pattern Mining Useful*

46 Loredana Afanasiev (UvA) *Querying XML: Benchmarks and Recursion*

2010

1 Matthijs van Leeuwen (UU) *Patterns that Matter*

2 Ingo Wassink (UT) *Work flows in Life Science*

3 Joost Geurts (CWI) *A Document Engineering Model and Processing Framework for Multimedia documents*

4 Olga Kulyk (UT) *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*

5 Claudia Hauff (UT) *Predicting the Effectiveness of Queries and Retrieval Systems*

6 Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*

7 Wim Fikkert (UT) *Gesture interaction at a Distance*

8 Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*

9 Hugo Kielman (UL) *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*

10 Rebecca Ong (UL) *Mobile Communication and Protection of Children*

11 Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*

12 Susan van den Braak (UU) *Sensemaking software for crime analysis*

13 Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*

14 Sander van Splunter (VUA) *Automated Web Service Reconfiguration*

15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*

16 Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*

17 Spyros Kotoulas (VUA) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*

18 Charlotte Gerritsen (VUA) *Caught in the Act: Investigating Crime by Agent-Based Simulation*

19 Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*

20 Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*

21 Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*

22 Michiel Hildebrand (CWI) *End-user Support for Access to Heterogeneous Linked Data*

23 Bas Steunebrink (UU) *The Logical Structure of Emotions*

24 Zulfiqar Ali Memon (VUA) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

25 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

26 Marten Voulon (UL) *Automatisch contracteren*

27 Arne Koopman (UU) *Characteristic Relational Patterns*

28 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*

29 Marieke van Erp (UvT) *Accessing Natural History: Discoveries in data cleaning, structuring, and retrieval*

30 Victor de Boer (UvA) *Ontology Enrichment from Heterogeneous Sources on the Web*

31 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

32 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

33 Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*

34 Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*

35 Jose Janssen (OU) *Paving the Way for Lifelong Learning: Facilitating competence development through a learning path specification*

36 Niels Lohmann (TUe) *Correctness of services and their composition*

37 Dirk Fahland (TUe) *From Scenarios to components*

38 Ghazanfar Farooq Siddiqui (VUA) *Integrative modeling of emotions in virtual agents*

39 Mark van Assem (VUA) *Converting and Integrating Vocabularies for the Semantic Web*

40 Guillaume Chaslot (UM) *Monte-Carlo Tree Search*

41 Sybren de Kinderen (VUA) *Needs-driven service bundling in a multi-supplier setting: the computational e3-service approach*

42 Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

43 Pieter Bellekens (TUe) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*

44 Vasilios Andrikopoulos (UvT) *A theory and model for the evolution of software services*

45 Vincent Pijpers (VUA) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*

46 Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*

47 Jahn-Takeshi Saito (UM) *Solving difficult game positions*

48 Bouke Huurnink (UvA) *Search in Audiovisual Broadcast Archives*

49 Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*

50 Peter-Paul van Maanen (VUA) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*

51 Edgar Meij (UvA) *Combining Concepts and Language Models for Information Access*

2011

1 Botond Cseke (RUN) *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*

2 Nick Tinnemeier (UU) *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*

3 Jan Martijn van der Werf (TUe) *Compositional Design and Verification of Component-Based Information Systems*

4 Hado van Hasselt (UU) *Insights in Reinforcement Learning: Formal analysis and empirical evaluation of temporal-difference*

5 Base van der Raadt (VUA) *Enterprise Architecture Coming of Age: Increasing the Performance of an Emerging Discipline*

6 Yiwen Wang (TUe) *Semantically-Enhanced Recommendations in Cultural Heritage*

7 Yujia Cao (UT) *Multimodal Information Presentation for High Load Human Computer Interaction*

8 Nieske Vergunst (UU) *BDI-based Generation of Robust Task-Oriented Dialogues*

9 Tim de Jong (OU) *Contextualised Mobile Media for Learning*

10 Bart Bogaert (UvT) *Cloud Content Contention*

11 Dhaval Vyas (UT) *Designing for Awareness: An Experience-focused HCI Perspective*

12 Carmen Bratosin (TUe) *Grid Architecture for Distributed Process Mining*

13 Xiaoyu Mao (UvT) *Airport under Control. Multiagent Scheduling for Airport Ground Handling*

14 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

15 Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

16 Maarten Schadd (UM) *Selective Search in Games of Different Complexity*

17 Jiyin He (UvA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18 Mark Ponsen (UM) *Strategic Decision-Making in complex games*

19 Ellen Rusman (OU) *The Mind ' s Eye on Personal Profiles*

20 Qing Gu (VUA) *Guiding service-oriented software engineering: A view-based approach*

21 Linda Terlouw (TUD) *Modularization and Specification of Service-Oriented Systems*

22 Junte Zhang (UvA) *System Evaluation of Archival Description and Access*

23 Wouter Weerkamp (UvA) *Finding People and their Utterances in Social Media*

24 Herwin van Welbergen (UT) *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*

25 Syed Waqar ul Qounain Jaffry (VUA) *Analysis and Validation of Models for Trust Dynamics*

26 Matthijs Aart Pontier (VUA) *Virtual Agents for Human Communication: Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*

27 Aniel Bhulai (VUA) *Dynamic website optimization through autonomous management of design patterns*

28 Rianne Kaptein (UvA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*

29 Faisal Kamiran (TUe) *Discrimination-aware Classification*

30 Egon van den Broek (UT) *Affective Signal Processing (ASP): Unraveling the mystery of emotions*

31 Ludo Waltman (EUR) *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*

32 Nees-Jan van Eck (EUR) *Methodological Advances in Bibliometric Mapping of Science*

33 Tom van der Weide (UU) *Arguing to Motivate Decisions*

34 Paolo Turrini (UU) *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*

35 Maaike Harbers (UU) *Explaining Agent Behavior in Virtual Training*

36 Erik van der Spek (UU) *Experiments in serious game design: a cognitive approach*

37 Adriana Burlutiu (RUN) *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*

38 Nyree Lemmens (UM) *Bee-inspired Distributed Optimization*

39 Joost Westra (UU) *Organizing Adaptation using Agents in Serious Games*

40 Viktor Clerc (VUA) *Architectural Knowledge Management in Global Software Development*

41 Luan Ibraimi (UT) *Cryptographically Enforced Distributed Data Access Control*

42 Michal Sindlar (UU) *Explaining Behavior through Mental State Attribution*

43 Henk van der Schuur (UU) *Process Improvement through Software Operation Knowledge*

44 Boris Reuderink (UT) *Robust Brain-Computer Interfaces*

45 Herman Stehouwer (UvT) *Statistical Language Models for Alternative Sequence Selection*

46 Beibei Hu (TUD) *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*

47 Azizi Bin Ab Aziz (VUA) *Exploring Computational Models for Intelligent Support of Persons with Depression*

48 Mark Ter Maat (UT) *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*

49 Andreea Niculescu (UT) *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

1 Terry Kakeeto (UvT) *Relationship Marketing for SMEs in Uganda*

2 Muhammad Umair (VUA) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*

3 Adam Vanya (VUA) *Supporting Architecture Evolution by Mining Software Repositories*

4 Jurriaan Souer (UU) *Development of Content Management System-based Web Applications*

5 Marijn Plomp (UU) *Maturing Interorganisational Information Systems*

6 Wolfgang Reinhardt (OU) *Awareness Support for Knowledge Workers in Research Networks*

7 Rianne van Lambalgen (VUA) *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*

8 Gerben de Vries (UvA) *Kernel Methods for Vessel Trajectories*

9 Ricardo Neisse (UT) *Trust and Privacy Management Support for Context-Aware Service Platforms*

10 David Smits (TUe) *Towards a Generic Distributed Adaptive Hypermedia Environment*

11 J. C. B. Rantham Prabhakara (TUe) *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*

12 Kees van der Sluijs (TUe) *Model Driven Design and Data Integration in Semantic Web Information Systems*

13 Suleman Shahid (UvT) *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*

14 Evgeny Knutov (TUe) *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*

15 Natalie van der Wal (VUA) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes*

16 Fiemke Both (VUA) *Helping people by understanding them: Ambient Agents supporting task execution and depression treatment*

17 Amal Elgammal (UvT) *Towards a Comprehensive Framework for Business Process Compliance*

18 Eltjo Poort (VUA) *Improving Solution Architecting Practices*

19 Helen Schonenberg (TUe) *What's Next? Operational Support for Business Process Execution*

20 Ali Bahramisharif (RUN) *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*

21 Roberto Cornacchia (TUD) *Querying Sparse Matrices for Information Retrieval*

22 Thijs Vis (UvT) *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*

23 Christian Muehl (UT) *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*

24 Laurens van der Werff (UT) *Evaluation of Noisy Transcripts for Spoken Document Retrieval*

25 Silja Eckartz (UT) *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*

26 Emile de Maat (UvA) *Making Sense of Legal Text*

27 Hayrettin Gurkok (UT) *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*

28 Nancy Pascall (UvT) *Engendering Technology Empowering Women*

29 Almer Tigelaar (UT) *Peer-to-Peer Information Retrieval*

30 Alina Pommeranz (TUD) *Designing Human-Centered Systems for Reflective Decision Making*

31 Emily Bagarukayo (RUN) *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*

32 Wietske Visser (TUD) *Qualitative multi-criteria preference representation and reasoning*

33 Rory Sie (OUN) *Coalitions in Cooperation Networks (COCOON)*

34 Pavol Jancura (RUN) *Evolutionary analysis in PPI networks and applications*

35 Evert Haasdijk (VUA) *Never Too Old To Learn: On-line Evolution of Controllers in Swarm- and Modular Robotics*

36 Denis Ssebugwawo (RUN) *Analysis and Evaluation of Collaborative Modeling Processes*

37 Agnes Nakakawa (RUN) *A Collaboration Process for Enterprise Architecture Creation*

38 Selmar Smit (VUA) *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*

39 Hassan Fatemi (UT) *Risk-aware design of value and coordination networks*

40 Agus Gunawan (UvT) *Information Access for SMEs in Indonesia*

41 Sebastian Kelle (OU) *Game Design Patterns for Learning*

42 Dominique Verpoorten (OU) *Reflection Amplifiers in self-regulated Learning*

43 Anna Tordai (VUA) *On Combining Alignment Techniques*

44 Benedikt Kratz (UvT) *A Model and Language for Business-aware Transactions*

45 Simon Carter (UvA) *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*

46 Manos Tsagkias (UvA) *Mining Social Media: Tracking Content and Predicting Behavior*

47 Jorn Bakker (TUe) *Handling Abrupt Changes in Evolving Time-series Data*

48 Michael Kaisers (UM) *Learning against Learning: Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*

49 Steven van Kervel (TUD) *Ontologogy driven Enterprise Information Systems Engineering*

50 Jeroen de Jong (TUD) *Heuristics in Dynamic Sceduling: a practical framework with a case study in elevator dispatching*

2013

1 Viorel Milea (EUR) *News Analytics for Financial Decision Support*

2 Erietta Liarou (CWI) *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*

3 Szymon Klarman (VUA) *Reasoning with Contexts in Description Logics*

4 Chetan Yadati (TUD) *Coordinating autonomous planning and scheduling*

5 Dulce Pumareja (UT) *Groupware Requirements Evolutions Patterns*

6 Romulo Goncalves (CWI) *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*

7 Giel van Lankveld (UvT) *Quantifying Individual Player Differences*

8 Robbert-Jan Merk (VUA) *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*

9 Fabio Gori (RUN) *Metagenomic Data Analysis: Computational Methods and Applications*

10 Jeewanie Jayasinghe Arachchige (UvT) *A Unified Modeling Framework for Service Design*

11 Evangelos Pournaras (TUD) *Multi-level Reconfigurable Self-organization in Overlay Services*

12 Marian Razavian (VUA) *Knowledge-driven Migration to Services*

13 Mohammad Safiri (UT) *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*

14 Jafar Tanha (UvA) *Ensemble Approaches to Semi-Supervised Learning Learning*

15 Daniel Hennes (UM) *Multiagent Learning: Dynamic Games and Applications*

16 Eric Kok (UU) *Exploring the practical benefits of argumentation in multi-agent deliberation*

17 Koen Kok (VUA) *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*

18 Jeroen Janssens (UvT) *Outlier Selection and One-Class Classification*

19 Renze Steenhuizen (TUD) *Coordinated Multi-Agent Planning and Scheduling*

20 Katja Hofmann (UvA) *Fast and Reliable Online Learning to Rank for Information Retrieval*

21 Sander Wubben (UvT) *Text-to-text generation by monolingual machine translation*

22 Tom Claassen (RUN) *Causal Discovery and Logic*

23 Patricio de Alencar Silva (UvT) *Value Activity Monitoring*

24 Haitham Bou Ammar (UM) *Automated Transfer in Reinforcement Learning*

25 Agnieszka Anna Latoszek-Berendsen (UM) *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*

26 Alireza Zarghami (UT) *Architectural Support for Dynamic Homecare Service Provisioning*

27 Mohammad Huq (UT) *Inference-based Framework Managing Data Provenance*

28 Frans van der Sluis (UT) *When Complexity becomes Interesting: An Inquiry into the Information eXperience*

29 Iwan de Kok (UT) *Listening Heads*

30 Joyce Nakatumba (TUe) *Resource-Aware Business Process Management: Analysis and Support*

31 Dinh Khoa Nguyen (UvT) *Blueprint Model and Language for Engineering Cloud Applications*

32 Kamakshi Rajagopal (OUN) *Networking For Learning: The role of Networking in a Lifelong Learner's Professional Development*

33 Qi Gao (TUD) *User Modeling and Personalization in the Microblogging Sphere*

34 Kien Tjin-Kam-Jet (UT) *Distributed Deep Web Search*

35 Abdallah El Ali (UvA) *Minimal Mobile Human Computer Interaction*

36 Than Lam Hoang (TUe) *Pattern Mining in Data Streams*

37 Dirk Börner (OUN) *Ambient Learning Displays*

38 Eelco den Heijer (VUA) *Autonomous Evolutionary Art*

39 Joop de Jong (TUD) *A Method for Enterprise Ontology based Design of Enterprise Information Systems*

40 Pim Nijssen (UM) *Monte-Carlo Tree Search for Multi-Player Games*

41 Jochem Liem (UvA) *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*

42 Léon Planken (TUD) *Algorithms for Simple Temporal Reasoning*

43 Marc Bron (UvA) *Exploration and Contextualization through Interaction and Concepts*

2014

1 Nicola Barile (UU) *Studies in Learning Monotone Models from Data*

2 Fiona Tuliyano (RUN) *Combining System Dynamics with a Domain Modeling Method*

3 Sergio Raul Duarte Torres (UT) *Information Retrieval for Children: Search Behavior and Solutions*

4 Hanna Jochmann-Mannak (UT) *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*

5 Jurriaan van Reijsen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*

6 Damian Tamburri (VUA) *Supporting Networked Software Development*

7 Arya Adriansyah (TUe) *Aligning Observed and Modeled Behavior*

8 Samur Araujo (TUD) *Data Integration over Distributed and Heterogeneous Data Endpoints*

9 Philip Jackson (UvT) *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language*

10 Ivan Salvador Razo Zapata (VUA) *Service Value Networks*

11 Janneke van der Zwaan (TUD) *An Empathic Virtual Buddy for Social Support*

12 Willem van Willigen (VUA) *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*

13 Arlette van Wissen (VUA) *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*

14 Yangyang Shi (TUD) *Language Models With Meta-information*

15 Natalya Mogles (VUA) *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*

16 Krystyna Milian (VUA) *Supporting trial recruitment and design by automatically interpreting eligibility criteria*

17 Kathrin Dentler (VUA) *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*

18 Mattijs Ghijsen (UvA) *Methods and Models for the Design and Study of Dynamic Agent Organizations*

19 Vinicius Ramos (TUe) *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*

20 Mena Habib (UT) *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*

21 Kassidy Clark (TUD) *Negotiation and Monitoring in Open Environments*

22 Marieke Peeters (UU) *Personalized Educational Games: Developing agent-supported scenario-based training*

23 Eleftherios Sidirourgos (UvA/CWI) *Space Efficient Indexes for the Big Data Era*

24 Davide Ceolin (VUA) *Trusting Semi-structured Web Data*

25 Martijn Lappenschaar (RUN) *New network models for the analysis of disease interaction*

26 Tim Baarslag (TUD) *What to Bid and When to Stop*

27 Rui Jorge Almeida (EUR) *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*

28 Anna Chmielowiec (VUA) *Decentralized k-Clique Matching*

29 Jaap Kabbedijk (UU) *Variability in Multi-Tenant Enterprise Software*

30 Peter de Cock (UvT) *Anticipating Criminal Behaviour*

31 Leo van Moergestel (UU) *Agent Technology in Agile Multiparallel Manufacturing and Product Support*

32 Naser Ayat (UvA) *On Entity Resolution in Probabilistic Data*

33 Tesfa Tegegne (RUN) *Service Discovery in eHealth*

34 Christina Manteli (VUA) *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems*

35 Joost van Ooijen (UU) *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*

36 Joos Buijs (TUe) *Flexible Evolutionary Algorithms for Mining Structured Process Models*

37 Maral Dadvar (UT) *Experts and Machines United Against Cyberbullying*

38 Danny Plass-Oude Bos (UT) *Making brain-computer interfaces better: improving usability through post-processing*

39 Jasmina Maric (UvT) *Web Communities, Immigration, and Social Capital*

40 Walter Omona (RUN) *A Framework for Knowledge Management Using ICT in Higher Education*

41 Frederic Hogenboom (EUR) *Automated Detection of Financial Events in News Text*

42 Carsten Eijckhof (CWI/TUD) *Contextual Multidimensional Relevance Models*

43 Kevin Vlaanderen (UU) *Supporting Process Improvement using Method Increments*

44 Paulien Meesters (UvT) *Intelligent Blauw: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden*

45 Birgit Schmitz (OUN) *Mobile Games for Learning: A Pattern-Based Approach*

46 Ke Tao (TUD) *Social Web Data Analytics: Relevance, Redundancy, Diversity*

47 Shangsong Liang (UvA) *Fusion and Diversification in Information Retrieval*

2015

1 Niels Netten (UvA) *Machine Learning for Relevance of Information in Crisis Response*

2 Faiza Bukhsh (UvT) *Smart auditing: Innovative Compliance Checking in Customs Controls*

3 Twan van Laarhoven (RUN) *Machine learning for network data*

4 Howard Spoelstra (OUN) *Collaborations in Open Learning Environments*

5 Christoph Bösch (UT) *Cryptographically Enforced Search Pattern Hiding*

6 Farideh Heidari (TUD) *Business Process Quality Computation: Computing Non-Functional Requirements to Improve Business Processes*

7 Maria-Hendrike Peetz (UvA) *Time-Aware Online Reputation Analysis*

8 Jie Jiang (TUD) *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*

9 Randy Klaassen (UT) *HCI Perspectives on Behavior Change Support Systems*

10 Henry Hermans (OUN) *OpenU: design of an integrated system to support lifelong learning*

11 Yongming Luo (TUe) *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*

12 Julie M. Birkholz (VUA) *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*

13 Giuseppe Procaccianti (VUA) *Energy-Efficient Software*

14 Bart van Straalen (UT) *A cognitive approach to modeling bad news conversations*

15 Klaas Andries de Graaf (VUA) *Ontology-based Software Architecture Documentation*

16 Changyun Wei (UT) *Cognitive Coordination for Cooperative Multi-Robot Teamwork*

17 André van Cleeff (UT) *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*

18 Holger Pirk (CWI) *Waste Not, Want Not!: Managing Relational Data in Asymmetric Memories*

19 Bernardo Tabuenca (OUN) *Ubiquitous Technology for Lifelong Learners*

20 Loïs Vanhée (UU) *Using Culture and Values to Support Flexible Coordination*

21 Sibren Fetter (OUN) *Using Peer-Support to Expand and Stabilize Online Learning*

22 Zhemin Zhu (UT) *Co-occurrence Rate Networks*

23 Luit Gazendam (VUA) *Cataloguer Support in Cultural Heritage*

24 Richard Berendsen (UvA) *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation*

25 Steven Woudenberg (UU) *Bayesian Tools for Early Disease Detection*

26 Alexander Hogenboom (EUR) *Sentiment Analysis of Text Guided by Semantics and Structure*

27 Sándor Héman (CWI) *Updating compressed column-stores*

28 Janet Bagorogoza (TiU) *Knowledge Management and High Performance: The Uganda Financial Institutions Model for HPO*

29 Hendrik Baier (UM) *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains*

30 Kiavash Bahreini (OUN) *Real-time Multimodal Emotion Recognition in E-Learning*

31 Yakup Koç (TUD) *On Robustness of Power Grids*

32 Jerome Gard (UL) *Corporate Venture Management in SMEs*

33 Frederik Schadd (UM) *Ontology Mapping with Auxiliary Resources*

34 Victor de Graaff (UT) *Geosocial Recommender Systems*

35 Junchao Xu (TUD) *Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction*

2016

1 Syed Saiden Abbas (RUN) *Recognition of Shapes by Humans and Machines*

2 Michiel Christiaan Meulendijk (UU) *Optimizing medication reviews through decision support: prescribing a better pill to swallow*

3 Maya Sappelli (RUN) *Knowledge Work in Context: User Centered Knowledge Worker Support*

4 Laurens Rietveld (VUA) *Publishing and Consuming Linked Data*

5 Evgeny Sherkhonov (UvA) *Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers*

6 Michel Wilson (TUD) *Robust scheduling in an uncertain environment*

7 Jeroen de Man (VUA) *Measuring and modeling negative emotions for virtual training*

8 Matje van de Camp (TiU) *A Link to the Past: Constructing Historical Social Networks from Unstructured Data*

9 Archana Nottamkandath (VUA) *Trusting Crowdsourced Information on Cultural Artefacts*

10 George Karafotias (VUA) *Parameter Control for Evolutionary Algorithms*

11 Anne Schuth (UvA) *Search Engines that Learn from Their Users*

12 Max Knobbout (UU) *Logics for Modelling and Verifying Normative Multi-Agent Systems*

13 Nana Baah Gyan (VU) *The Web, Speech Technologies and Rural Development in West Africa: An ICT4D Approach*

14 Ravi Khadka (UU) *Revisiting Legacy Software System Modernization*

15 Steffen Michels (RUN) *Hybrid Probabilistic Logics: Theoretical Aspects, Algorithms and Experiments*

16 Guangliang Li (UvA) *Socially Intelligent Autonomous Agents that Learn from Human Reward*

17 Berend Weel (VUA) *Towards Embodied Evolution of Robot Organisms*

18 Albert Meroño Peñuela (VUA) *Refining Statistical Data on the Web*

19 Julia Efremova (TUe) *Mining Social Structures from Genealogical Data*

20 Daan Odijk (UvA) *Context & Semantics in News & Web Search*

21 Alejandro Moreno Célleri (UT) *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground*

22 Grace Lewis (VU) *Software Architecture Strategies for Cyber-Foraging Systems*

23 Fei Cai (UvA) *Query Auto Completion in Information Retrieval*

24 Brend Wanders (UT) *Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach*

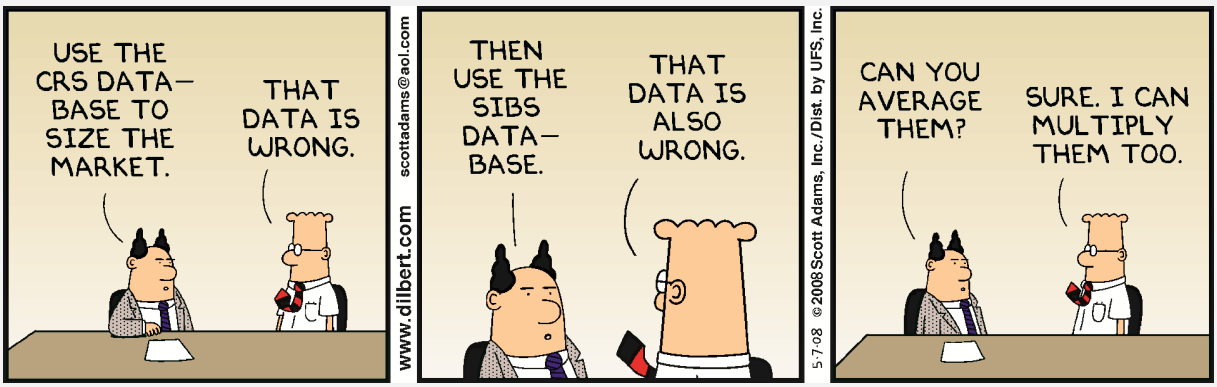
25 Julia Kiseleva (TUe) *Using Contextual Information to Understand Searching and Browsing Behavior*

26 Dilhan Thilakarathne (VU) *In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains*

27 Wen Li (TUD) *Understanding Geo-spatial Information on Social Media*

28 Mingxin Zhang (TUD) *Large-scale Agent-based Social Simulation - A study on epidemic prediction and control*

29 Nicolas Höning (CWI/TUD) *Peak reduction in decentralised electricity systems - Markets and prices for flexible planning*



ISBN 978-90-365-4110-7