Crowd-sourcing Domain Semantics to Guide Mapping Selection in Dataspaces

MSc Progress Report

Iliada Eleftheriou

Supervisor:
Dr Suzanne Embury

2012 - 2013
ABSTRACT

Data integration systems require full integration of sources before providing any services, causing significant up-front costs, often higher than the cost of system ownership. Dataspaces propose a new pay-as-you-go approach for integrating sources which supports on-demand integration and low initialisation costs. A dataspace uses techniques to find relationships among the sources and incrementally refines these relationships by enabling incremental collection of domain semantics needed to integrate sources.

One of the challenges that dataspaces face is the improvement of the selection of mappings (the relationships between sources). While different dataspace proposals investigate different mechanisms to collect domain semantics, currently the most dominant input provided to assist mapping selection is user feedback. Although this kind of domain semantic can be very useful to the dataspaces, it is often absent and expensive to collect.

This project explores a different way of gathering domain semantics to guide mapping selection in dataspaces, called crowdsourcing systems. Crowdsourcing systems enlist a crowd of users to collectively solve a problem. The project aims to investigate whether crowdsourcing is a cost-efficient and reliable source for gathering domain semantics to guide mapping selection in dataspaces.

This report introduces and analyses the above hypothesis, provides the background concepts of the project, describes the progress made so far in terms of the deliverables and project plan and finally discusses the future work to be completed.
ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Suzanne M. Embury for her tremendous help, support and guidance along with the huge enthusiasm and availability.

I also thank Ruhaila Maskatr for dedicating her time to attend the meetings and for providing valuable feedback.

Last but not least, special thanks to my family not only for providing the funding to complete this MSc but for giving me endless love, support and encouragement.
Table of Contents

Abstract ................................................................................................................................................... 1
Acknowledgements ................................................................................................................................. 2
Table of Contents .................................................................................................................................... 3
List of Figures .......................................................................................................................................... 5
List of Tables ........................................................................................................................................... 5
Glossary................................................................................................................................................... 6
Chapter 1: Introduction .................................................................................................................... 7
  1.1 Aim and Objectives .................................................................................................................... 7
  1.2 Project Context .......................................................................................................................... 8
  1.3 Deliverables ................................................................................................................................ 8
  1.4 Report Outline ............................................................................................................................ 9
Chapter 2: Background - Data Integration Systems ........................................................................ 10
  2.1 Data Integration Systems ......................................................................................................... 10
    2.1.1 Architecture .................................................................................................................. 10
    2.1.2 Global As View .............................................................................................................. 11
    2.1.3 Schema Matching .......................................................................................................... 12
    2.1.4 Summary ....................................................................................................................... 13
  2.2 Dataspaces ............................................................................................................................... 14
    2.2.1 Aim ................................................................................................................................ 14
    2.2.2 Process .......................................................................................................................... 14
  2.3 Summary .................................................................................................................................. 17
Chapter 3: Background - Crowdsourcing Systems .......................................................................... 18
  3.1 Definition .................................................................................................................................. 18
  3.2 Crowdsourcing Platforms ......................................................................................................... 18
  3.3 Challenges ................................................................................................................................ 19
  3.4 Amazon's Mechanical Turk ....................................................................................................... 19
    A. Begin with a project and break it into tasks ......................................................................... 21
    B. Publish HITs to the marketplace ........................................................................................... 21
    C. Workers accept assignments ............................................................................................... 22
    D. Workers submit assignments for review .............................................................................. 22
    E. Approve or reject assignments ............................................................................................. 22
  3.5 Summary .................................................................................................................................. 22
Chapter 4: Interim Product ............................................................................................................. 23
4.1 Deliverables .............................................................................................................................. 23
4.2 Forms of feedback .................................................................................................................... 24
4.3 Crowdsourcing properties ....................................................................................................... 25
4.4 Forms of Domain Knowledge ................................................................................................. 27
4.5 Forms of domain knowledge & crowdsourcing properties ..................................................... 27
4.6 Experiment Design .................................................................................................................. 28
  System Architecture .................................................................................................................. 28
  Part 1: Dataspace ...................................................................................................................... 29
  Part 2: CrowdSpace ................................................................................................................... 29
  Part 3: Amazon’s Mechanical Turk .......................................................................................... 30
4.7 Summary .................................................................................................................................. 30
Chapter 5: Research Methods & Project Plan ............................................................................... 31
  5.1 Research methodology ......................................................................................................... 31
  5.2 Evaluation Plan ..................................................................................................................... 31
  5.3 Project Plan ............................................................................................................................ 32
    5.3.1 Milestones & Deliverables ............................................................................................ 33
    5.3.2 Progress so far ............................................................................................................... 33
    5.3.3 Future work ................................................................................................................... 34
  5.4 Summary .................................................................................................................................. 35
Chapter 6: Conclusion .................................................................................................................... 36
Bibliography .......................................................................................................................................... 37
Appendix A – Query Processing Example ..................................................................................... 39
Appendix B – HIT Examples .......................................................................................................... 42
Appendix C – Project Plan ............................................................................................................... 0
LIST OF FIGURES

Figure 1: Global and local schemas ................................................................. 11
Figure 2: Dataspace Lifecycle ........................................................................ 15
Figure 3: MT process ..................................................................................... 21
Figure 4: HIT #1 ............................................................................................ 21
Figure 5: HIT #2 ............................................................................................ 21
Figure 6: CrowdSpace architecture ............................................................... 28

LIST OF TABLES

Table 2: Categorisation of deliverables......................................................... 23
Table 3: Deliverable #1 - Forms of feedback ................................................ 25
Table 4: Deliverable #2 - Crowdsourcing properties .................................... 26
Table 5: HIT details ....................................................................................... 30
<table>
<thead>
<tr>
<th><strong>Glossary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assignment:</strong></td>
</tr>
<tr>
<td><strong>Crowdsourcing:</strong></td>
</tr>
<tr>
<td><strong>Dataspaces:</strong></td>
</tr>
<tr>
<td><strong>Domain semantics:</strong></td>
</tr>
<tr>
<td><strong>GAV:</strong></td>
</tr>
<tr>
<td><strong>Global schema:</strong></td>
</tr>
<tr>
<td><strong>HIT group:</strong></td>
</tr>
<tr>
<td><strong>HIT:</strong></td>
</tr>
<tr>
<td><strong>Matchings:</strong></td>
</tr>
<tr>
<td><strong>MT:</strong></td>
</tr>
<tr>
<td><strong>Requester:</strong></td>
</tr>
<tr>
<td><strong>Semantic mappings:</strong></td>
</tr>
<tr>
<td><strong>Sources:</strong></td>
</tr>
<tr>
<td><strong>Worker:</strong></td>
</tr>
</tbody>
</table>
Chapter 1: INTRODUCTION

Overview

Dataspaces are a new approach to integrating data sources that overcomes the up-front costs of traditional data integration systems, caused by the need for full integration of the sources before useful queries can be answered in such systems. This is achieved by providing incremental integration based on the current needs of the user. In particular, when a user poses a query, the dataspace firstly creates a plausible integration of the sources required to return the result. If the quality of the integration is found to be insufficient for the user’s current purposes, the dataspace will refine the initial combination of parts of the sources by collecting domain semantics (information regarding the domain of interest). Although domain semantics can be very useful to the improvement of the quality of a dataspace, they are often very expensive to collect.

This MSc project will investigate whether forms of domain semantics (also referred as domain knowledge) needed to improve mapping selection can be gathered cost-effectively using crowdsourcing. Crowdsourcing systems enlist a crowd of users to collectively solve a problem often tasks difficult for the computer to achieve.

The report will investigate different forms of domain semantics that can be gathered cost-effectively through crowdsourcing along with potential uses by the dataspace. Existing forms of information gathered by crowdsourcing and forms of domain knowledge needed in mapping selection in dataspaces will be investigated to propose some candidate forms for further exploration. The most promising one will be selected for an experimental study to ensure whether reliable information of the chosen form can be gathered through crowdsourcing.

This chapter will give an introduction to the research hypothesis tested in this project and describes the aim and objectives, the context and scope of the project along with the deliverables of the final project.

1.1 Aim and Objectives

The aim of the project is to determine whether crowdsourcing can be a suitable, cost-effective and reliable source of useful domain information to be used in dataspaces. The approach to be followed to achieve this aim is to seek a single example of a kind of domain knowledge that is useful in carrying out mapping selection in dataspaces and that can be gathered cost-effectively by crowdsourcing.

The objectives of the project are:

1. Collate a broad sample of forms of domain information that can be used by a dataspace when performing mapping selection.
2. Review the literature to identify properties of domain knowledge that make it a suitable candidate for gathering through crowdsourcing.
3. Determine which of the forms of domain knowledge identified under objective 1 may be amenable to collection by crowdsourcing (based upon the properties identified under objective 2). From this subset, I will select one for further analysis.
4. Determine, through an empirical evaluation, whether the hypothesis that the form of domain knowledge selected under objective 3, truly is suitable for collection by crowdsourcing, under the control of a dataspace.

1.2 Project Context

Data Integration is the combination of data residing at different sources providing the user with a unified view of these data with aim of supporting aggregation, exploration and mining [1]. Designing a data integration system requires the creation of a global schema which provides a virtual view of the sources, along with the semantic mappings between the schemas of the data sources and the global schema [1]. When the user asks a query over the global schema, the query is reformulated into a set of queries that can be evaluated by the individual sources, based on the semantic mappings. The results from these individual queries are then recombined, and translated into the terms/structures used in the global schema. This classical approach to data integration requires significant up-front and on-going effort since it demands full integration of sources before providing any services.

Dataspaces are a data management abstraction aiming to reduce the up-front costs required to integrate data sources, by incrementally specifying schema mappings using a pay-as-you-go approach [2, 6]. The pay-as-you-go approach creates a collection of (often incorrect) semantic mappings, which are then improved over time through interaction with the end user. The user interacts with the system by providing information about the correctness of the results of the queries that were evaluated using the generated mappings [6].

However, there are lots of different kinds of domain information that are needed, in combination, to fully correct the selection of mappings. Current research on dataspaces has studied only a small fraction. Such domain knowledge is needed to improve the selection of mappings for use in query processing but is often expensive to collect [5].

‘On-Demand Data Integration: Dataspaces by Refinement’ is a research project on-going in our school to create a tool set for assessing the quality of an integration, diagnosing failures in integration and gathering the domain semantics needed for improvement, while keeping the costs low for the user [7]. This MSc project will contribute towards this research, by investigating the forms of domain knowledge that can be gathered by crowdsourcing and how the information collected can be potentially used by the dataspace.

1.3 Deliverables

The final results from this project will consist of the following deliverables:

\[ a. \quad \text{A review of the literature on forms of feedback used by dataspaces, and the ways in which they are gathered.} \]

Dataspaces require feedback from the user in order to refine the schema mappings it uses, to improve the integration of the sources and therefore the correction of the query results. A review of the literature will identify the different forms of feedback provided by the user, studied so far and the ways in which dataspaces proposals currently gather this feedback.
b. A review of the literature on crowd-sourcing, focussing on crowd-sourcing on domain-specific knowledge.

Crowdsourcing systems help to solve a variety of problems. Therefore a deliverable of the project is a review of the literature on the properties that domain knowledge must have in order to be gathered effectively by crowdsourcing.

c. A set of proposals for forms of domain knowledge that can assist in mapping selection.

The third deliverable is a set of different forms of domain knowledge that can be used by the dataspaces to select the candidate mappings that will be used for query evaluation.

d. A categorisation of the forms of domain knowledge in terms of the degree to which they are amenable to collection by crowdsourcing.

The fourth deliverable is an analysis of the documented proposals as to whether they possess the characteristics and properties required by crowdsourcing systems.

e. An experiment design for one of the proposals that seems to be most promising.

Produce an experiment design for the selected form of domain knowledge to investigate whether it will be suitable for crowdsourcing. The experiment will be launched through the Mechanical Turk of Amazon (crowdsourcing application).

f. Results of the experiment.

Collect responses from the crowd through Mechanical Turk and evaluate them to determine whether they can be used in mapping selection by the dataspace producing a set of recommendations of domain semantics that can be gathered through crowdsourcing.

1.4 Report Outline

The report is structured in the following sections:

Chapter 2 – Background – Data Integration Systems: This section describes the wider context of the project starting by exploring the concept and challenges of data integration systems, introducing dataspaces, a new approach to overcome those challenges.

Chapter 3 – Background – Crowdsourcing systems: This chapter introduces the concept of crowdsourcing as a different way of collecting domain semantics to assist mapping selection in dataspaces and provides the key points to consider when creating a crowdsourcing system.

Chapter 4 – Project progress: This segment of the report presents the progress made so far in terms of the deliverables of the project.

Chapter 5 – Research methodology and Project plan: This chapter discusses the methodology used to conduct literature review, describes the project plan including milestones, deadlines, progress so far and future work.

Chapter 6 – Conclusion: This section summarises the key points stated in the report, provides the challenges phased so far.
Chapter 2: BACKGROUND - DATA INTEGRATION SYSTEMS

Overview

‘Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone’ [1]. Database systems are not only growing in size but they have also expanded in scope and complexity. Data integration facilitates the conversion of all these big data into knowledge to provide a competitive advantage to companies and organisations. However, traditional data integration systems require that a full description of the relationships between the sources be created at the start of the project, causing high up-front costs — often higher than the cost of system ownership.

This chapter aims to provide general background on data integration systems emphasising traditional approaches to query processing in data integration systems and their challenges. It also introduces a new approach, the dataspace, which aims to provide the benefits of traditional systems while overcoming (some of) their weaknesses.

2.1 Data Integration Systems

The concept of integrating data has been the focus of study by many research groups for over 20 years. The objective of data integration is the combination of data residing in different sources, providing the user with a unified view of these data with aim of supporting aggregation, exploration and mining of data [2–4]. By facilitating a unified view of the sources, it gives the illusion of accessing a single database, while in fact data can be stored and managed in different geographic locations by different set of technologies and languages.

One of the most important benefits of data integration systems is that they enable end users to focus on what they want to achieve instead of how to retrieve the results from the isolated sources. The system finds the necessary data sources, interacts with them to request data and combines the resulting data from multiple sources to answer the queries posed by the users.

2.1.1 Architecture

Data Integration systems are characterised by an architecture containing a global schema and a set of different sources. The global schema (also referred as the mediated or integration schema) describes the data residing in the sources and provides a virtual view of one system containing relevant aspects of the domain of interest represented in a format that is convenient for the purposes for which the integration system is required. The global schema may not contain all the relations and attributes of the sources, only the ones that the user is interested in. The sources consist of the real data. The schemas of the sources are called local schemas and are related to the global schema using semantic mappings. The semantic mappings model the relation between the data sources and the global schema, [1, 2] by specifying how data of the sources can be transformed and/or combined into data of the global schema [5]. Several approaches modelling these mappings have been explored. Our work is based on one of these approaches called Global-As-View (GAV) since it is the one that the majority of current systems implement.
The following diagram illustrates an example of data integration system for the “Hogwarts University”\(^1\) containing 3 different sources, the students’ database, the professors’ database and the supervisions database containing information on professors supervising students. Each source is an independent database, illustrated using the database symbol, and has an autonomous local schema. The square boxes represent the different schemas and the arrow-lines the mappings between the local and global schema for each source.

![Diagram](image_url)

**Figure 1: Global and local schemas**

### 2.1.2 Global As View

In the GAV approach the global schema is defined in terms of mappings over the data sources. More specifically, for each relation (table) \( R \) of the global schema, the integration team must define a query \( q \) over the relations of the data sources which specifies how to retrieve the tuples of \( R \) from the sources [3].

Using the example of “Hogwarts University” introduced in section 2.1.1 and illustrated in figure 1, suppose that \( I = <G, S, M> \) is the data integration system, where \( G \) is the global schema, \( S \) the set of sources and \( M \) the set of mappings. We have 3 data sources:

\[
S = \{ S_1, S_2, S_3 \}
\]

- \( S_1 : \text{StudentDB}(sId, sFName, sLName, dob, level) \)
- \( S_2 : \text{ProfessorDB}(pId, pName, area) \)

---

\(^1\) Hogwarts University’ is a work of fiction. Names, characters, places and incidents either are products of the author’s imagination (J. K. Rowling) or are used fictitiously by the author of this report. Any resemblance to actual events or locales or persons, living or dead, is entirely coincidental.
The global schema of the system is:

\[ G = \{ \text{UGstudent(ugId, ugName), Faculty(profId, profName), Supervisions(ugId, profId)} \} \]

The relationship between the sources and the global schema is given by the set of mappings:

\[ M = \{ M_1, M_2, M_3 \} \]

\[ M_1 : \text{UGstudent(ugId, ugName) \rightarrow \{ sId, sFName + " " + sLName | StudentDB(sId, sFName, sLName, dob, year) and year <= 4 \} } \]

\[ M_2 : \text{Faculty(profId, profName) \rightarrow \{ pId, pName | ProfessorDB(pId, pName, area) \} } \]

\[ M_3 : \text{Supervisions(ugId, profId)} \rightarrow \{ sId, pId | SupervisionsDB(sId, pId) and StudentDB(sId, _, _, _, year) and year <= 4 \} \]

In GAV, extending the system is difficult since the new source can impact several different elements of the global schema which will need to be refined. However, query processing is relatively simple since the relations of the global schema are expressed in terms of the relations of the sources and query reformulation requires only unfolding the mappings [1, 4]. The following section describes schema matching and explains how these mappings would be derived in a typical integration project.

### 2.1.3 Schema Matching

Building a data integration system requires the specification of a global schema and the description of how the autonomous sources’ schemas relate to it.

A data integration schema contains numerous independent sources to be integrated. Having a global schema, sources must be mapped with the global schema. The relationship between sources and global schema is represented with matches and mappings. Matches indicate similar attributes between sources.

There are numerous tools for matching different schemas [7, 10–12]. The result of those tools can be 1-1 matches or complex matches. An example of 1-1 matching is to match location with address (e.g. table “loc” matches with table “add”) whereas a complex matching can be the matching of name with the concatenation of first and second name (attribute “name” matches with attribute concat(fname, lname)).

A matching tool produces candidate matchings between attributes of the global schema and similar attributes of the sources. The list of matchings produced is ranked in decreasing order based on confidence. An example list of candidate 1-1 matchings retrieved from the ‘Hogwarts University’ schemas in section 2.1.1 is:

- \( \text{ugId} = \text{sId} \)
- \( \text{ugId} = \text{pId} \)
The list of candidates is then reviewed by the developer/programmer to identify the correct match. However, there is an extremely high possibility of the correct matching not to be included in the list because of the difficulty of the task the matching tools face\(^2\) coupled with the large size of source schemas of real world data integration systems.

The developer/programmer must manually look all the candidate matchings for each element of the global schema and find a precise query that will return the exact results that the user needs. This is achieved using schema and instance match information or any related documentation if exist [11], making it a labour-intensive and error-prone process [7, 8]. In addition, users often don’t fully understand the semantics of the local schemas. However, data residing in the local schemas present data quality issues and data heterogeneities causing the presence of results that we don’t expect at the result set. An example of data quality issue is that the details of a postgraduate student to be stored in the ‘ugStudents’ table for historical reasons, or because the student used to be an undergraduate and now is a postgraduate student, or because of a data entry error occurred.

### 2.1.4 Summary

Data integration systems require full semantic integration before providing any services. Therefore, configuring and maintaining a data integration application requires significant upfront effort in order to create the global schema and semantic mappings connecting data sources to the global schema [13]. As a result, data integration systems are considered to be in a high-cost, high-quality position of the data access spectrum and less effective for several rapidly changing data sources [14].

However, many application contexts do not require initial full integration to facilitate useful services and provide results, proposing the use of a new pay-as-you-go approach to integration. The following section describes the benefits of this new approach, while introducing the concept of dataspaces.

---

\(^2\) Textual similarity tests on names will not reveal the strong semantic link between the names “student” and “undergraduate”, for example. Special knowledge of the domain is needed.
2.2 Dataspaces

Overview

Recently, a new approach to integrating data has been studied that aims to reduce the up-front and on-going costs required to set up a data integration system. This new kind of data integration system is called a dataspace. Dataspaces aim to reduce costs by incrementally specifying schema mappings based on the current needs of the user. The quality of the integration is mainly based on the correctness of schema mappings making mapping selection a main challenge of dataspaces.

2.2.1 Aim

The aim of dataspaces is to simplify integrated access to distributed data by providing benefits of traditional data integration while reducing up-front and ongoing costs. They provide an opportunity for incremental refinement of the integration, enabling the user to pay the cost of the integration they need at any time and to avoid costs that don’t bring any immediate return [14].

2.2.2 Process

As in a data integration system, the first step of a dataspace is to initialise the system by identifying the data sources and integrate them to produce a global schema and a set of relationships between the global schema and source schemas [15]. Once initialisation is completed the next phase is the usage of the dataspace. The user poses queries in terms of the global schema; however as discussed earlier in section 2.1.1, data are stored in sources using their own local schemas. In the usage phase the dataspace expands the query of the user into several queries over the sources, translates the sub-queries into the query language used in each source and finally evaluates the queries [16]. The usage phase is followed by the improvement phase which enables incremental refinement of the integration. All three phases are implemented automatically by the dataspace. The user only indicates the sources to be integrated.

The following figure summarises the 3 phases of the dataspace lifecycle as described in [16] providing the parts related with this MSc investigation. The arrows represent the flow of process of the dataspace.
The following sub-sections describe each phase of the dataspace lifecycle in more details focusing on the steps and concepts that fall under the scope of this report.

**Phase 1: Initialisation**

The initialisation phase consists of the identification and integration of data sources. Schema matching is relatively the same process as the one traditional data integration systems follow described in section 2.1.3. However, dataspaces handle all the tasks of schema matching, user only specifies which sources to be integrated. Then the dataspace extracts the schemas of each source and imports them into its repository. Once the schemas are imported, the dataspace runs the relevant schema matching tools to produce the candidate list of matchings which is saved in the dataspace's metadata repository. The process of schema matching may take several hours for real scale systems.

Because of the high up-front costs and huge workload required to find the correct matching, dataspaces do not expect to find the optimal correct matching but takes the top $k$ matchings from the list of candidates ($k$ is an integer number and differs in the different dataspace research proposals). Consequently, dataspaces enable incremental refinement of the matchings at the improvement phase.
Having identified the matchings with the highest confidence scores, the dataspace must find a precise way to map the tables of the sources with each table in global schema: i.e., to automatically create mappings from the matchings. There are numerous tools generating mappings automatically used by dataspaces described in [6 - 9, 11]. The main difference between dataspaces and traditional data integration systems is that dataspaces must create multiple mappings for each table of the global schema, because it cannot know at this stage which one meets the requirements of the user precisely.

**Phase 2: Usage**

The usage phase consists of the reformulation of the query posed over the global schema into queries over the sources, the translation of those queries into the language used in each source and finally the evaluation of the queries.

Query reformulation is achieved by the query processor with the help of the mappings based on an unfolding strategy. In the traditional GAV approach, since each global schema table has exactly one mapping, a query posed on the Global Schema can be easily unfolded into sub-queries over the sources. However, in a dataspace, the query must be reformulated using *all* the candidate mappings generated in initialisation phase (or a subset containing the ones with the highest confidence scores) since the single correct mapping is not yet known. Hence, the query is reformulated by producing the union of sub-queries produced by each combination of the selected mappings [16]. Then the reformulated queries are optimised and any sub-queries on sources using different language are being translated and evaluated. The query evaluator returns the result to the user which often contains results that the user would not expect to see because of large sets of mappings, complex queries and the absent knowledge regarding correctness of mappings.

An example implementing query reformulation, translation and evaluation was retrieved from [16] and is presented in Appendix A.

**Phase 3: Improvement**

The improvement phase improves the integration done in the initialisation phase based on the quality of the result set produced at the end of the usage phase. As mentioned in the usage phase, dataspace uses *all* the mappings to reformulate the query which may cause undesirable consequences [16]. If there is a large number of mappings of poor quality there is high risk of taking long time for the query to be evaluated and will result to a set of tuples containing lots of wrong answers. Hence the dataspace must select the mappings to be used when evaluating the query. This process is called *mapping selection*. Different dataspace proposals investigate different mechanisms to gain additional knowledge in order to improve the selection of mappings and therefore the quality of the integration. The most dominant mechanism enables the user to provide feedback. In [17, 18] dataspace asks the user to give feedback on the construction of mappings. However this approach requires the user to have a good understanding of schemas and mappings [16]. A different approach asks the user to give feedback on the result set of the query. For example the user can annotate tuples that were not expected to be retrieved and/or tuples that were expected to be retrieved and they were not. A further discussion on forms of feedback is given in section 0.
2.3 Summary

There are lots of different kinds of domain information that are needed, in combination, to fully correct the selection of mappings. Current research on dataspaces has studied only a small fraction. This kind of domain semantic can be very useful to the dataspaces specifically for improving the selection of mappings; however it is often expensive to collect.

The following chapter introduces the concept of crowdsourcing as a different way of collecting domain semantics.
Chapter 3: BACKGROUND - CROWDSOURCING SYSTEMS

Overview

Although machines provide tremendous benefits to humanity they have limitations. Respectively, humans have their strengths and weaknesses, they are good and bad at certain tasks, just like computers. Crowdsourcing systems enable the interaction of humans and computers by enlisting human dynamics to solve tasks difficult to be achieved by a computer.

Consequently, this chapter introduces the concept of crowdsourcing as a different way of collecting domain semantics to assist mapping selection in dataspaces. Also, this chapter explores crowdsourcing platforms and systems, their challenges. It summarises the key points to consider when creating a crowdsourcing system and finally investigates the process of crowdsourcing a job using Amazon’s Mechanical Turk platform.

3.1 Definition

Crowdsourcing systems enlist a crowd of users to explicitly or implicitly collaborate to build a long-lasting artefact or to solve a general purpose non-trivial problem that is beneficial to the community or the system owners [19].

3.2 Crowdsourcing Platforms

Recently, many crowdsourcing platforms that recruit users to contribute to different tasks have been built. Some examples are:

- Mechanical Turk (https://www.mturk.com/mturk/welcome)
- Turkit (http://groups.csail.mit.edu/uid/turkit/)
- Mob4hire (http://www.mob4hire.com/)
- uTest (http://www.utest.com/)
- Freelancer (http://www.freelancer.com/)
- eLance (https://www.elance.com/)
- oDesk (https://www.odesk.com/)
- Guru (http://www.guru.com/index.aspx)
- TopCoder (http://www.topcoder.com/)

These systems provide platforms for building crowdsourcing systems in different domains easily and quickly. However, there is a set of fundamental challenges that must be addressed before start creating a crowdsourcing system.
3.3 Challenges

Crowdsourcing systems in general, face four main challenges [19].

1. How to recruit and retain participants?
2. What contributions can participants make?
3. How to combine participants’ contributions to solve the target problem?
4. How to evaluate participants and their contributions?

In particular, in terms of this MSc project a crowdsourcing system will be designed and implemented that will gather domain semantics to assist in mapping selection in dataspaces. This kind of crowdsourcing system requires the consideration not only of the above four main challenges but of the following points as well.

1. What questions to ask the participants?
2. How to pose those questions?

Those challenges aim to ask the participants questions that are easy for humans but difficult for the machines to answer [11], while the result obtained from asking them would benefit dataspaces in respect with mapping selection. In addition, questions must be posed in natural language and therefore may inherit ambiguities since they can be interpreted in different ways [20].

3. How to visualise posed questions?

The design and layout of the interface and the structure of both the questions and answers can affect the speed and accuracy of the participants [20].

4. How many responses are needed?

Crowdsourcing systems must cope with the amount of participants available and willing to answer the posed questions. Some participants may be specialised in particular kinds of questions or prefer to work with certain people because of their reputation of providing well-defined tasks or paying appositely.

5. How to evaluate participants’ reliability and quality of their answers?

The responses of the participants often vary in quality since online communities may contain noisy participants that are ignorant and/or malicious. Answers of such participants can have a negative impact to the survey and hence cannot be trusted.

The above list contains only the most important set of challenges that must be considered and addressed when designing a crowdsourcing system. Further details on the different properties of crowdsourcing systems are described in section 4.3.

3.4 Amazon’s Mechanical Turk

The Amazon Mechanical Turk (MT) is a leading crowdsourcing platform which provides an Internet marketplace. Requesters of MT offer tasks that require human intelligence to solve problems that are impossible or very expensive to be answered correctly by a computer [20]. Workers (also known
as Providers or Turkers) can select a task to work on and receive a monetary payment set by the Requester [21].

- **Requester:**

  Requesters can post tasks that can be viewed only by workers that meet a set of criteria, called qualifications [22]. Tests can be set in order to ensure that workers fulfill the appropriate qualifications (examples of tasks requiring qualifications can be found in appendix B). Once a task is completed by a worker and approved to be valid, the requester pays the worker the predefined reward of at least $0.01 (reward rarely exceeds $1.00 per task) [20, 21].

- **Worker:**

  Workers can participate from all around the world to answer the tasks posed by requesters. Answers provided by workers can be accepted or rejected by the requester. Good quality work is rewarded with bonuses while requesters might refuse to pay for poor quality work and the worker may be blocked from completing future tasks [19, 21]. Each worker maintain a reputation based on the quality of their responses. Requesters can use the reputation of a worker to assess the reliability of his/her work.

- **HIT (Human Intelligent Task):**

  The tasks posted by the requesters are called HIT (Human Intelligent Tasks). Typically, HITs do not require any training and they are simple enough to be completed in less than a minute [20, 21]. Examples of HITs can be found in Appendix B.

- **Assignment:**

  Each HIT can be completed several times by different workers. Each answer is called assignment [20]. A worker can work in at most one assignment meaning that can only solve a HIT once. This constraint enables the requester to collect responses from multiple workers for a single HIT.

- **HIT Group:**

  Similar HITs are grouped together automatically by MT into an HIT Group. HITs are grouped based on their title, requester, description and reward. Workers can search HITs and choose one based on HIT Groups [20]. An example of HIT Group is presented below.

**Process/Walkthrough**

The following figure illustrates the steps of a requester posting HITs in MT, workers solving the tasks and finally requesters accept or decline answers to form the solution of the problem. The figure is
retrieved from the MT website [25]. The following sections describe each step in more details.

**A. Begin with a project and break it into tasks**

This step is about defining the goals of the project to be crowdsourced. For example, a project could be to identify text out of images for authentication reasons, like the examples illustrated below. The project may include lots of images to be solved. Each image will be a different HIT. Then, all those HITs will be grouped together as a HIT Group. Hence, the HIT Group will contain all the pictures that need to be solved. A worker can submit an answer for a HIT (a picture) once. The submission the worker is called assignment. Therefore there is only one assignment per worker per HIT.

**B. Publish HITs to the marketplace**

The requester must assign clear instructions for the group of HITs, showing some results so the workers can understand and provide precise and high quality responses. For example above the two answers would be “don’t be late” and “no fuel surcharges” respectively. Then the requester must provide a reward per HIT, a set of requirements/qualifications that workers must satisfy in order to complete the HIT in this group and the maximum number of assignments for each HIT.
C. Workers accept assignments

The workers can now view the HIT groups and check whether they pass all the qualifications to complete the HITs in each group, the reward and a short description of the HITs in the group as shown in the figure below.

D. Workers submit assignments for review

When a HIT reaches its maximum number of assignments which means different workers have completed the same task up to the specified number, the certain HIT becomes invisible and is not available to workers.

E. Approve or reject assignments

Once the workers submit their assignments per HIT, the requester can review them and decide whether they reflect the initial goal. The requester can approve or reject a submission. Each accepted submission is paid with reward. In the case where a worker has completed a task accurately and in good quality, the requester has also the choice to add a bonus to his pre-defined reward.

3.5 Summary

"Given the success of current crowdsourcing systems, we expect that this emerging field will grow rapidly [19]."

Crowdsourcing systems proved to be a successful way of tackling difficult for the machines tasks and their application is expected to grow rapidly in the following years [19]. Although they present a set of challenges that need to be addressed, crowdsourcing systems provide numerous benefits to a wide range of domains. In particular, this chapter identified the main challenges to be considered when creating a crowdsourcing system to collect domain semantics to assist in mapping selection.

The following chapter describes the progress made so far in terms of the deliverables of the project.
Chapter 4: Interim Product

Overview

This chapter describes the interim product produced so far which contributes towards the final product of the project. The final product will contain the research regarding the experiment and the creation of a system to evaluate the research hypothesis of the project. The following sections discuss the progress achieved so far on both parts of the final product. The first part is represented as a set of tables containing information, details and comparisons regarding the deliverables of the project.

4.1 Deliverables

The final product derived from this MSc project will contain the 6 deliverables described in section 1.3 and summarized in the following table. These 6 deliverables are categorized into 4 different parts. The first part contains reviews of the literature to extract essential information to be used in the following deliverables. The second part proposes and categorises possible proposals for forms of knowledge that can assist mapping selection of dataspaces. The third part contains an experiment design of the most promising proposal to evaluate whether it will provide reliable results cost-effectively, if crowdsourced. Finally the fourth part consists of the results of the experiment and a set of recommendations of domain $k$ that can be gathered through crowdsourcing.

The following sections describe the progress made on each part of the final product. Each part is presented in a different way in order to get the most out of them. The first part is presented using tables containing all the essential information gathered, while the second part combines, contradicts and critiques the tables produced in part 1. Part 3 presents and discusses the experiment design.

<table>
<thead>
<tr>
<th>No</th>
<th>Deliverable</th>
<th>Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A review of the literature on forms of feedback used by dataspaces, and the ways in which they are gathered.</td>
<td>1 – Literature Review</td>
</tr>
<tr>
<td>2</td>
<td>A review of the literature on crowd-sourcing, focussing on crowd-sourcing on domain-specific knowledge.</td>
<td>1 – Literature Review</td>
</tr>
<tr>
<td>3</td>
<td>A set of proposals for forms of domain knowledge that can assist in mapping selection.</td>
<td>2 – Proposals</td>
</tr>
<tr>
<td>4</td>
<td>A categorisation of the forms of domain knowledge in terms of the degree to which they are amenable to collection by crowdsourcing.</td>
<td>2 – Proposals</td>
</tr>
<tr>
<td>5</td>
<td>An experiment design for one of the proposals that seems to be most promising.</td>
<td>3 – Experiment</td>
</tr>
<tr>
<td>6</td>
<td>Results of the experiment – set of recommendations.</td>
<td>4 – Results</td>
</tr>
</tbody>
</table>

Table 1: Categorisation of deliverables
4.2 Forms of feedback

*Deliverable #1: A review of the literature on forms of feedback used by dataspaces, and the ways in which they are gathered.*

User feedback can be requested and used in a number of different phases in a dataspace lifecycle. For example, users can comment on the quality of the results of the queries by annotating the tuples that were not meant to be in the result set. By doing so dataspaces can use user feedback to improve the mappings used to produce the undesirable tuples.

However, user feedback can assist not only in the improvement of mappings but in the formulation and maintenance of matchings, mappings and integration schema, in the reformulation of queries, query results or in the ranking of the results. Any feedback obtained can be reused in any other stage of a dataspace lifecycle, for improvement, not only in the stage in which it was acquired [14].

There are 2 kinds of user providing feedback to the dataspaces:

1. **Domain Experts**: people familiar with one or more of the domains covered by the information to be integrated.
2. **Database Experts**: people with a good understanding of the source schemas, the integration schema, the relations between them and the technical concepts that underlie them.

User feedback can be asked for:

1. providing any of the following elements:
   a. Matchings
   b. Mappings
   c. Global schema
   d. Query results
2. refining any of the elements stated above,
3. annotating above elements with quality measures. The results of a query may be annotated with scores of some form, or they may be ranked.

The following table implements the first deliverable of the project by describing the forms of feedback used by different dataspace proposals along with the source expected to get the feedback and the way it was gathered.

<table>
<thead>
<tr>
<th>No</th>
<th>Dataspace proposal</th>
<th>Forms of feedback</th>
<th>Description/Problem</th>
<th>Expected from:</th>
<th>How is gathered?</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>DS Toolkit [5]</td>
<td>TP = a given tuple was expected in the answer</td>
<td>Refining the query result set: A set of tuples were expected to appear at the query result.</td>
<td>User</td>
<td>[TBA]</td>
<td>1. Query result set is refined. 2. Mappings are improved.</td>
</tr>
</tbody>
</table>
The above collection of forms of feedback used by dataspaces will be used to determine what kind of information is expected to be retrieved by the workers of the crowdsourcing system.

### 4.3 Crowdsourcing properties

Crowdsourcing systems help to solve a variety of problems and therefore can be classified along many dimensions. The second deliverable of the project is the identification of the properties that domain knowledge must have in order to be gathered effectively by crowdsourcing.

*Deliverable #2: A review of the literature on crowd-sourcing, focussing on crowd-sourcing on domain-specific knowledge.*

The following table identifies the different properties that a crowdsourcing system could have. This table will be used to assess suitability of the different forms of domain knowledge, identified in section 0, to be crowdsourced. The rows of this table will represent the column headings of the
fourth deliverable table which categorises forms of domain knowledge in terms of the degree to
which they are amenable to collection by crowdsourcing.

<table>
<thead>
<tr>
<th>No</th>
<th>Property</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Nature of collaboration</td>
<td>Explicit OR Implicit</td>
<td>1. Explicit systems let users collaborate explicitly. 2. Users collaborate implicitly to solve a problem.</td>
</tr>
<tr>
<td>2.3</td>
<td>Recruiting users</td>
<td>a. Require users</td>
<td>a. Require users to make contributions if we have the authority to do so.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b. Pay users</td>
<td>b. Mechanical Turk for example provides a way to pay users on the Web to help with a task.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>c. Volunteers</td>
<td>c. Free and easy to execute, and hence is most popular but inefficient due to the quality.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>d. Pay for service</td>
<td>d. To require the users of system “X” to pay for using “X”, by contributing to a CS system “Y”. E.g. Solving a puzzle in a website proving that you are a human being and not a spam machine.</td>
</tr>
<tr>
<td>2.4</td>
<td>Combining user input</td>
<td>1. No combination</td>
<td>1. E.g. combine numeric ratings using simple formulas.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Link contributions</td>
<td>2. E.g. use in networking systems to form a social network graph.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Combine contributions</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>Evaluating user input</td>
<td>[TBA]</td>
<td>[TBA]</td>
</tr>
<tr>
<td>2.6</td>
<td>Degree of manual effort</td>
<td>[TBA]</td>
<td>[TBA]</td>
</tr>
<tr>
<td>2.7</td>
<td>Role of human users</td>
<td>1. Slaves</td>
<td>1. Humans help solve the problem in a divide and conquer fashion, to minimise the resources of the owners.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Perspective providers</td>
<td>2. Humans contribute different perspectives, which when combined often produce a better solution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Content providers</td>
<td>3. Humans contribute self-generated content (e.g. Videos on YouTube)</td>
</tr>
<tr>
<td>2.8</td>
<td>Standalone OR Piggyback</td>
<td>-</td>
<td>[TBA]</td>
</tr>
</tbody>
</table>

Table 3: Deliverable #2 - Crowdsourcing properties

Examples of general domain knowledge that are currently being collected through crowdsourcing are the followings. The following examples are retrieved from Amazon’s Mechanical Turk [21].

a. Translate a paragraph from English to Spanish.
b. Select the correct spelling for these search terms.
c. Find the item number of a product in this image.
d. Rate the search results for a set of keywords.
e. Are these two products the same?
f. Choose the most appropriate category for products.
g. Categorize the tone of this article.

A further investigation on forms of domain knowledge that can be effectively collected by crowdsourcing will happen in the future.
4.4 Forms of Domain Knowledge

The third deliverable is a set of different forms of domain knowledge that can be used by the dataspaces to select the candidate mappings that will be used for query evaluation.

*Deliverable #3: A set of proposals for forms of domain knowledge that can assist in mapping selection.*

The following table identifies aspects where different domain knowledge can be used to evaluate mapping selection. This table will be used in deliverable 4 to explore which of the below domain knowledge can be collected through crowdsourcing. The rows of this table represent the candidate proposals of deliverable 4.

<table>
<thead>
<tr>
<th>No</th>
<th>Domain Knowledge</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Missing information</td>
<td>Use of domain knowledge to find missing information of the sources.</td>
<td>The first name of Dr. Smith.</td>
</tr>
<tr>
<td>3.2</td>
<td>Incorrect information</td>
<td>Use of domain knowledge</td>
<td>[TBA]</td>
</tr>
<tr>
<td>3.3</td>
<td>Domain conflicts</td>
<td>[TBA]</td>
<td>[TBA]</td>
</tr>
<tr>
<td>3.4</td>
<td>Semantic mismatches among sources</td>
<td>[TBA]</td>
<td>[TBA]</td>
</tr>
<tr>
<td>3.5</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This deliverable is one of the most challenging to be completed since the need of different ways to assist mapping selection was recently identified and therefore only a limited number of proposals investigated ways to overcome it.

4.5 Forms of domain knowledge & crowdsourcing properties

The fourth deliverable is an analysis of the documented proposals as to whether they possess the characteristics and properties required by crowdsourcing systems.

*Deliverable #4: A categorisation of the forms of domain knowledge in terms of the degree to which they are amenable to collection by crowdsourcing.*

Forms of Domain Knowledge that can assist in mapping selection with crowdsourcing properties:

- **Rows**: forms of **domain knowledge** that assist dataspaces in mapping selection (deliverable #3).
- **Columns**: **properties** that make domain knowledge amenable to crowdsourcing (deliverable #4).

This deliverable will be formulated using the two previous deliverables, hence it will be created after the completion of deliverable #2 & #3.
4.6 Experiment Design

After selecting the most promising proposal of a domain knowledge that would be useful in mapping selection, the next step is to evaluate whether gathering it through crowdsourcing would indeed be an efficient, reliable and most importantly cost-efficient way.

**Deliverable #5: An experiment design for one of the proposals that seems to be most promising.**

Deliverable 5 evaluates whether the selected proposal satisfies the project hypothesis. The experiment will be run through the Mechanical Turk crowdsourcing system of Amazon. The progress achieved so far for this deliverable contains the architecture of the system to be built and an analysis on the flow of data transferred between the system and Mechanical Turk.

**System Architecture**

The following figure illustrates the initial architecture of the system to be built. The system architecture is divided into 3 parts. The part illustrated on the left hand side of the figure is the dataspace system. The part in the right is Amazon’s Mechanical Turk, and finally the part in the middle is the system that will be developed as part of this project indicated with red colour. Since it is a component that will link the dataspace system with the crowdsourcing system, we have given it the name ‘CrowdSpace’.

![CrowdSpace architecture](image)

The following sections describe the flow of data transferred between the crowdSpace Mediator and Mechanical Turk and provide some initial, simple ideas on how crowdSpace can interact with the dataspace which will be refined in the future.
**Part 1: Dataspase**

Dataspase system requires a user who will supply the queries to be answered. The user can also provide feedback to assist the dataspace when needed. During the maintenance and improvement phase of the dataspace (see section 2.2.2), the matchings, mappings and global schema are being refined either because there was a change in the underlying sources or the quality of the result set is not sufficient. During refinement, dataspace can either reuse results of previous integration tasks [14], or/and ask the user to provide some feedback. CrowdSpace mediator will provide an additional option to the dataspace to assist refining the selection of mappings. This component will be investigated in the future after completing deliverable #4 in section 4.5.

**Part 2: CrowdSpace**

The crowdSpace component will receive a request from the dataspace containing the problem to be solved. CrowdSpace will then decide whether the problem is suitable for being crowdsourced, to reply to the request either positively (can handle the job) or negatively.

The following paragraphs provide a very simple example of abstract interaction between dataspace and crowdSpace since further investigation will be achieved in the future.

An example of request sent by the dataspace is:

“Is the name ‘Fred’ male or female or unisex?”

CrowdSpace will then decide whether the problem is suitable for being crowdsourced by analysing the request in terms of the following:

- **Description/Title:** “Is the name”
- **Input type Value:** String
- **Set of possible answers:** “male”, “female”, “unisex”, “don’t know”
- **Qualifications:** default: none
- **Reward:** default: $0.01
- **Type of HIT:** Categorisation

The CrowdSpace represents the data as it was analysed and accepted to be crowdsourced back to the DataSpace system. The user can then view/edit this form and let the CrowdSpace revalidate it to proceed on publication. In the case where the edited form or the initial request cannot pass the properties of crowdsourcing systems identified in section 4.3, it will reject the request, displaying to the user why.

Other examples of requests could be:

1. **Type Equality:** e.g. “Is ‘student’ the same with ‘UGstudent’?”
2. **Categorisation:** e.g. “Is ‘079036734895’ a telephone number?”
3. **Data Collection:** e.g. “What is the website of ‘Hogwarts University’?”
4. **Image moderation:** e.g. “Does the image contain violence?”
5. **Video moderation:** e.g. “Does the video contain the Beatles?”
If the crowdSpace decides that the request can be handled, then the content of the request must be converted into a form that can be manipulated by Amazon’s Mechanical Turk (AMT). The following table summarises the information that crowdSpace must provide to Mechanical Turk for each HIT (Human Intelligence Task).

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The title of the HIT.</td>
</tr>
<tr>
<td>2</td>
<td>The description of the HIT.</td>
</tr>
<tr>
<td>3</td>
<td>The reward that workers will receive after successfully deliver an approved HIT.</td>
</tr>
<tr>
<td>4</td>
<td>Requirements that workers must satisfy in order to complete the HIT. (optional)</td>
</tr>
<tr>
<td>5</td>
<td>Number of assignments required to be completed.</td>
</tr>
<tr>
<td>6</td>
<td>Quality control methods.</td>
</tr>
<tr>
<td>7</td>
<td>Test Criteria in order to approve or decline an assignment.</td>
</tr>
</tbody>
</table>

**Table 4: HIT details**

### Part 3: Amazon’s Mechanical Turk

CrowdSpace sends a request to MT containing all the necessary information to post a HIT, using the API provided by MT [27]. Data needed to create/manipulate an HIT and all relevant MT features are identified below. The list below provides the methods needed to be implemented for achieving the following functionality:

1. **Crowdsource a valid HIT on the MT marketplace.**
   In order to do that the method `createHIT` can be used along with the `title, description, reward, duration and maxAssignments` (The number of times the HIT can be accepted) parameters which are required by MT to be published for the workers.

2. **Retrieve results from previous assignments.**
   `GetAssignmentsForHIT` returns the results of all assignments of a specific HIT that have been accepted and completed successfully by workers.

3. **Approve or reject an assignment.**
   Calling the `ApproveAssignment` method enables the requester to either approve or reject the results of assignments; therefore approving/negating the payment reward to workers.

4. **Remove/extend a HIT from the marketplace.**
   The requester can make a HIT unavailable/inactive or he can remove/extend it using the `ForceExpireHIT` and `DisableHIT/ExtendHIT` interfaces respectively.

### 4.7 Summary

This chapter described the interim product produced so far in terms of the deliverables of the project which contributes towards the final product. However, some of the above sections are not yet fully completed. Once they are finished, we will select the most promising proposal of form of domain knowledge which fulfills the requirements to be crowdsourced and check whether can be efficiently collected through crowdsourcing.

The following chapter describes the research methods used to implement the first part of deliverables of the project, the project plan followed to achieve the numerous tasks and finally the future work.
Chapter 5: Research Methods & Project Plan

Overview

This chapter describes the research methods used to conduct literature review. It also discusses the project plan that was followed, milestones, deadlines, progress so far and future work.

5.1 Research methodology

Research is a “structured enquiry that utilizes acceptable scientific methodology to solve problems and create new knowledge that is generally applicable” [28]. Having a methodology to organise research is essential to stay focus and avoid additional costs in terms of time and unnecessary functionality. Followed are the research methods used during the project so far.

1. Background reading

The first step of the project was to identify the keywords underlying in the title of the project and establish background research to explore the different concepts and understand the wider context of data integration systems, dataspaces and crowdsourcing.

2. Research

Once the basic background context was investigated, the next step was to explore research articles to firstly derive the hypothesis of the project and then to find important challenges and proposals to overcome those challenges if exist.

3. Mini lectures

Dataspaces and crowdsourcing are new areas of research and consequently there is only a limited number of articles related with those areas and most of them describe specific parts of these concepts. Hence in order to get a complete picture I had mini-lecture sessions with Ruhaila Maskatr and Klitos Christodoulou, both Ph.D students working on the ‘On-Demand Data Integration: Dataspaces by Refinement’ research project on-going in our school.

4. Literature review

Because of the nature of this project being a research experiment, a subset of the deliverables is literature review aiming to extract proposed approaches and numerous proposals, criticise them, and combine them to get an overview of the approaches used.

5.2 Evaluation Plan

The evaluation plan consists of three phases:

- The first phase will evaluate the aim & objectives of the project.
- The second phase will evaluate all the functional and non-functional requirements retrieved from the objectives of the project.
- Finally, the third phase will evaluate the results of the system to be built.
This experiment can have 3 different outcomes.

a) Positive outcome: Chosen domain semantics were gathered cost-effectively and reliably by crowdsourcing, while the results will assist mapping selection of dataspace.

b) Partially positive outcome:
   a. Chosen domain semantics were gathered cost-effectively but not reliably.
   b. Chosen domain semantics were gathered reliably but not cost-effectively.
   c. Chosen domain semantics were gathered cost-effectively and reliably but results were not helpful to the dataspace.
   d. Chosen domain semantics were gathered reliably but not cost-effectively and were helpful to the dataspace.

c) Negative outcome: Chosen domain semantics were gathered neither cost-effectively nor reliably by crowdsourcing, and the results were not helpful to the dataspace.

5.3 Project Plan

The following figure contains a Gantt chart illustrating the plan of this project. The left part of the diagram consists of the deliverables broken down into smaller tasks. The right section contains the timeline and the performance so far. A larger, more readable version of the plan can be found in Appendix B.
The project is divided into four phases. The first phase contains the background reading and the implementation of the 4 first deliverables which are described in section 1.3 and are presented in chapter 5, Project Progress. The second phase consists of the selection of the most promising proposal to experiment. Phase 3 contains the design and implementation of the experiment along with collection and analysis of responses and final conclusions. The final phase contains the writing of the thesis. The following sections describe the milestones and deliverables of the project, progress so far and future work.

5.3.1 Milestones & Deliverables

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone/Deadline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26/4/2013</td>
<td>Finish phase 1</td>
<td>1. Finish background reading on data integration systems, dataspaces and crowdsourcing systems. 2. Finish literature review required to complete the first 4 deliverables.</td>
</tr>
<tr>
<td>7/6/2013</td>
<td>Finish phase 2</td>
<td>1. Review MT. 2. Choose the most promising domain semantic.</td>
</tr>
<tr>
<td>19/7/2013</td>
<td>Launch experiment</td>
<td>Design, implement and launch crowdSpace system.</td>
</tr>
<tr>
<td>16/8/2013</td>
<td>Finish evaluation part</td>
<td>Collect responses from MT and evaluate hypothesis.</td>
</tr>
</tbody>
</table>

5.3.2 Progress so far

<table>
<thead>
<tr>
<th>#</th>
<th>Tasks</th>
<th>Percentage Completed</th>
<th>Mitigated Risks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Understand the process of integrating two or more sources.</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>Actual</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Explore the challenges of integrating heterogeneous sources.</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Identify a set of forms of domain knowledge that can assist in mapping selection in dataspaces.</td>
<td>100%</td>
<td>Challenging deliverable: since only a limited number of proposals investigated.</td>
</tr>
<tr>
<td>4</td>
<td>Explore the properties that make a problem suitable for crowd-sourcing.</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Determine for each form of knowledge whether it fulfils each property.</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>6</td>
<td>Write background report.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>7</td>
<td>Choose the most suitable form of domain knowledge that can be gathered with crowdsourcing.</td>
<td>Yes (Yes/No)</td>
<td>Not yet</td>
</tr>
<tr>
<td>8</td>
<td>Review how Mechanical Turk works.</td>
<td>50%</td>
<td>70%</td>
</tr>
<tr>
<td>9</td>
<td>Design an experiment to evaluate the chosen domain knowledge.</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>10</td>
<td>Implement the program that will send the jobs to Mechanical Turk and automatically get responses.</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>11</td>
<td>Launch the experiment to Mechanical Turk.</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>12</td>
<td>Collect all the user responses.</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>13</td>
<td>Analyse results.</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>14</td>
<td>Write Thesis.</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

**Part 2: Proposing a solution**

**Part 3: Evaluation of solution**

**Part 4: Writing Thesis**

**5.3.3 Future work**

This section provides the future work to be completed by the submission of the final thesis in September.

- Finish literature review on Mechanical Turk, find the final set of data that is needed to be crowdsourced.
- Continue investigating DataSpace on the data to be automatically filtered out and decide how the user will interact with the Crowdspace system if needed.
- Decide if the amount of data is enough to consider this as a solution for crowdsourcing.
- Write tests for accepting several types of requests from the DataSpace to the CrowdSpace.
- Implement a simulation of DataSpace (plug in) and create several requests containing valid and invalid data for crowdsourcing.
- Continue to the implementation of initialising the HITs on the market place in MT.
- Run the tests and evaluate the results, then refine.
- Implement further features based on refinement.
5.4 Summary

This chapter described the research methods used to conduct literature review, discussed the project plan that was followed, the milestones and deadlines to be met, the progress so far and the future work.

The next chapter gives the challenges of the project so far and concludes the report.
Chapter 6: **Conclusion**

Researching new areas of study proved to be a very challenging process since the only source for extracting knowledge is the numerous articles written by different people, describing different proposals and using different notations. In addition, having to investigate such a specific concept turned to be a challenge not to get lost between all the different ideas, implementations, extra details and to stay focus on the scope of the project. However practicing achieves knowledge and having the guidance from my supervisor I learnt that research is a way of thinking.

This report introduced and analysed the hypothesis of whether crowdsourcing can be a suitable, cost-effective and reliable source of useful domain information to be used in dataspaces. It provided the background concepts of the project, described the progress made so far in terms of the deliverables and project plan and finally discussed the future work to be completed.


3. A. Y. Halevy, “Logic-Based Techniques in Data Integration.”


5. N. W. Paton and S. M. Embury, “Feedback-Based Annotation, Selection and Refinement of Schema Mappings for Dataspaces * Categories and Subject Descriptors.”


The following example is retrieved from the ‘DSToolkit: An architecture for flexible dataspace management’ [16].

1. The user imports the sources and defines the global schema.

Example source schemas

2. Dataspaces match schemas.

Matches between source schemas
3. Dataspace find schematic correspondences.

4. Dataspace generates integration schema.
   
   country(name, code, capital, area, population)
   city(name, country, population, longitude, latitude)
   language(country, name, percentage)

   Merged schema $s_{m2}$

5. Dataspace finds mappings.

   $map_1 = \langle s_{m2}.country, select o.name, o.code, o.capital, o.area, o.population from s_1.country o >$
   
   $map_2 = \langle s_{m2}.city, select c.name, c.country, c.population, c.longitude, c.latitude from s_1.city c >$
   
   $map_3 = \langle s_{m2}.country, select o.name, o.code, o.capital, o.area, o.population from s_2.countries o >$
   
   $map_4 = \langle s_{m2}.city, select c.name, c.country, c.population, c.longitude, c.latitude$
   from s_2.cities c, s_2.location l$
   where c.name = l.city and c.country = l.country$
   
   $map_5 = \langle s_{m2}.country, select o.country.name as name, o.code, o.capital, o.area, o.population$
   from s_3.country o$
   
   $map_6 = \langle s_{m2}.language, select l.country, l.language.name as name, l.percentage from s_3.country.language l >$

   Mappings between $s_{m2}$ and $s_1$, $s_2$, $s_3$
6. User poses query and annotates query results.

<table>
<thead>
<tr>
<th>name</th>
<th>code</th>
<th>capital</th>
<th>area</th>
<th>population</th>
<th>expected</th>
<th>not expected</th>
<th>mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>F</td>
<td>Paris</td>
<td>547030</td>
<td>58317450</td>
<td>✔</td>
<td></td>
<td>map1, map5</td>
</tr>
<tr>
<td>Turkey</td>
<td>TR</td>
<td>Ankara</td>
<td>780580</td>
<td>62484478</td>
<td>✔</td>
<td></td>
<td>map1, map5</td>
</tr>
<tr>
<td>Italy</td>
<td>I</td>
<td>Rome</td>
<td>301230</td>
<td>57460274</td>
<td>✔</td>
<td></td>
<td>map1, map5</td>
</tr>
<tr>
<td>Tunisia</td>
<td>TN</td>
<td>Tunis</td>
<td>163610</td>
<td>9019657</td>
<td>✔</td>
<td></td>
<td>map3, map5</td>
</tr>
<tr>
<td>Morocco</td>
<td>MA</td>
<td>Rabat</td>
<td>446550</td>
<td>29779156</td>
<td>✔</td>
<td></td>
<td>map3, map5</td>
</tr>
<tr>
<td>Algeria</td>
<td>DZ</td>
<td>Algier</td>
<td>2.38174e+06</td>
<td>29183032</td>
<td>✔</td>
<td></td>
<td>map3, map5</td>
</tr>
</tbody>
</table>
**APPENDIX B – HIT EXAMPLES**

This Appendix contains examples of HIT (Human Intelligent Task) retrieved from the Amazon’s crowdsourcing platform, Mechanical Turk.

1. List of HITs posted.

<table>
<thead>
<tr>
<th>Requester</th>
<th>HIT Expiration Date</th>
<th>Time Allowed</th>
<th>Reward</th>
<th>HITs Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon requester Inc. - Browse classification</td>
<td>May 16, 2013</td>
<td>10 minutes</td>
<td>$0.06</td>
<td>2221</td>
</tr>
<tr>
<td>Amazon requester Inc.</td>
<td>May 25, 2013</td>
<td>1 hour 46 minutes</td>
<td>$0.00</td>
<td>2892</td>
</tr>
<tr>
<td>Cloudflow</td>
<td>May 6, 2014</td>
<td>60 minutes</td>
<td>$2.25</td>
<td>2607</td>
</tr>
<tr>
<td>Checklists</td>
<td>May 13, 2013</td>
<td>60 minutes</td>
<td>$0.01</td>
<td>2471</td>
</tr>
<tr>
<td>Checklists</td>
<td>May 8, 2013</td>
<td>1 day 13 hours</td>
<td>$0.07</td>
<td>2587</td>
</tr>
<tr>
<td>Limited HLP Group</td>
<td>May 12, 2013</td>
<td>60 minutes</td>
<td>$0.03</td>
<td>1000</td>
</tr>
<tr>
<td>IT Services</td>
<td>May 13, 2013</td>
<td>30 minutes</td>
<td>$0.04</td>
<td>1728</td>
</tr>
<tr>
<td>Amazon requester Inc.</td>
<td>May 25, 2013</td>
<td>1 hour 46 minutes</td>
<td>$0.00</td>
<td>1600</td>
</tr>
<tr>
<td>Amazon requester Inc.</td>
<td>May 26, 2013</td>
<td>1 hour 46 minutes</td>
<td>$0.00</td>
<td>1600</td>
</tr>
</tbody>
</table>
2. Describe an image.

Provide 5 tags for this image.

Instructions:
- You must provide 5 tags for this image.
- Each tag must be a single word.
- No tag can be longer than 25 characters.
- The tags must describe the image, the contents of the image, or some relevant context.

Please ignore the "yukee M" watermark.
Please visit this page for examples: [http://mturk.yukee-m.com/add_keywords.html](http://mturk.yukee-m.com/add_keywords.html)

<table>
<thead>
<tr>
<th>Tag 1:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag 2:</td>
<td></td>
</tr>
<tr>
<td>Tag 3:</td>
<td></td>
</tr>
<tr>
<td>Tag 4:</td>
<td></td>
</tr>
<tr>
<td>Tag 5:</td>
<td></td>
</tr>
</tbody>
</table>
3. Identify unrelated image results regarding the query: North Korea

4. Write questions based on keywords.
5. Extract data from receipt.

![Image of a receipt with text: Please extract data from this image of a receipt. Instructions: Enter text exactly as it appears. If the photo is of a receipt but is missing data, enter your best guess. You will be awarded a bonus for long receipts of approximately 0.03€ for every ten items on the receipt. Enter numbers without any additional characters (1.99€ instead of $1.99 or 1.99 lb). If items are not shown on the receipt, enter 'Purchase' as the item, and the total as the price. Over 95% of our hits should be marked as possible. If your % of hits marked as possible gets too high, we will warn and then block you. This is an example of how receipts should be entered.

1. Can you complete this task? *
   - Yes.
   - No, the receipt is not legible.
   - No, the image is not a receipt.
   - No, other reason: [ ]

2. What is the name of the company on the receipt? *
   - WALMART

Location information powered by Foursquare

3. Where and when was this purchase made? (if shown)
### Appendix C – Project Plan

**Crowd-sourcing Domain Semantics to Guide Mapping Selection in Dataspaces – Project Plan**

Ilida Eleftheriou - 7425045  
MSc in Advanced Computer Science and IT Management

<table>
<thead>
<tr>
<th>#</th>
<th>Tasks</th>
<th>Start (Week)</th>
<th>Duration (Weeks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Understand the process of integrating two or more sources.</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Explore the challenges of integrating heterogeneous sources.</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Identify a set of forms of domain knowledge that can assist in mapping selection in dataspaces.</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Explore the properties that make a problem suitable for crowd-sourcing.</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Determine for each form of knowledge whether it fulfils each property.</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Write background report.</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td><strong>Part 2: Proposing a solution</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Choose the most suitable form of domain knowledge that can be gathered with crowd-sourcing.</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Review how Mechanical Turk works.</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td><strong>Part 3: Evaluation of solution</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design an experiment to evaluate the chosen domain knowledge.</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Implement the program that will send the jobs to Mechanical Turk and automatically get responses.</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>Launch the experiment to Mechanical Turk.</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Collect all the user responses.</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>Analyse results.</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td><strong>Part 4: Writing Thesis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Write Thesis.</td>
<td>27</td>
<td>5</td>
</tr>
</tbody>
</table>

**Note:**  
- **Milestone**  
- **Deadline**  
- 16/8: Finish evaluation part  
- 6/9: Submit thesis  
- 26/4: Finish part 1  
- 10/5: Submit background report  
- 7/6: Finish part 2  
- 19/7: Launch Experiment