A Survey on Incrementally Improving Dataspace Systems Using User Feedback

Avinash Kumar[1], Srishti Agarwal[2], Mr. Mrityunjay Singh[3]

[1,2]Students of M.TECH (CSE), SRM University, India
[3]Assistant Professor in SRM University, India

Abstract: **- Dataspace is a prominent research area in heterogeneous data management. A dataspace system provides a new data integration approach which integrates its data incrementally and improves its performance with time. Because a dataspace system deals a variety of heterogeneous data, and there is various kind of heterogeneity present among the data at schema level, data level and query level. These heterogeneities can be categories as structural heterogeneity, syntactic heterogeneity, and semantic heterogeneity. In order to improve the performance of system there is a requirement of addressing the semantic heterogeneity at various levels. In literature, we have found various algorithms based on the user feedbacks. In this paper, we survey the existing algorithms for modeling the semantic heterogeneity in dataspace and incrementally improving the performance of a dataspace system.**

Keywords: **- Data integration, Dataspaces, semantic heterogeneity, user-feedback.**

I. INTRODUCTION

At present, dataspace technology is the most crucial research area in data management community. This concept was introduced by M. Franklin et al in 2005 in which they acknowledge the drawback of existing data management system such as database system, traditional data integration system etc [1]. These systems require lots of upfront efforts in defining the schemas, and finding the semantic relationship among them. These activities are time consuming and error pruning. The dataspace system provides an evaluation over the traditional data integration systems. M. Franklin has argued that the dataspace system requires fewer efforts, integrates the data without human involvement, and provides a best effort answer to the users with time [11]. Unlike a traditional data integration system, dataspace system does not require a precise semantic integration among the data. The data and semantic relationships among them are involved over time. The applications of a dataspace system include structured data management in web, personal information management (PIM), scientific data management, and so on. A dataspace system has distinguished properties from the existing data management systems [12].

Dataspaces are very much different from databases. There are different areas in which great advancement has been done in the last few years. Some are:

- schema matchings
- mappings
- Heterogeneities
- Reference reconciliation
- Information extraction etc.

The major contributions on Dataspaces have been done in the "pay-as-you-go approach", which is a technique for determining how users can help to improve the Dataspace systems by providing their feedbacks to the system. In this work, we have survey the existing techniques of modeling the semantic heterogeneity on Dataspace and improving the performance of the system with time. We have presented the existing techniques in chrononical order in the following section. The existing techniques are:

- Pay-as-you-go User Feedback for Dataspace Systems [2]
- Feedback-Based Annotation, Selection and Refinement of Schema Mappings for Dataspaces [3]
- Building Data Integration Systems: A Mass Collaboration Approach[5]
- Pay-As-You-Go Mapping Selection in Dataspaces[6]

- User Feedback as a First Class Citizen in Information Integration Systems [7]
- Soliciting User Feedback in a Dataspace System [8]
- Incrementally improving dataspaces based on user feedback [9]

The rest of paper is organized as follows: Section 2 present the various existing techniques based on user feedback. In section 3, we present a theoretical comparison among the various existing techniques. Finally, we have concluded our work in section 4.

II. EXISTING TECHNIQUES

2.1 Pay-as-you-go User Feedback for Dataspace Systems [2]

In this approach, the authors have created the candidate matches from the predefined schemas and then ask for user feedback just to confirm these matches. There is a confliction between different schemas in the dataspaces in which same data are used in the records but with different names or we can say different tuple names. For example, suppose, we have two different schemas and each schema have some records. Each record contains a tuple for name but in one record, the tuple_name is "Name" and in other record, the tuple name is "Student_Name". Both of the tuples will result in similar data. So, it is better to remove this type of confliction from the schemas for simplicity and easy performance. This can be done by the use of semantic heterogeneity. By modeling Semantic ***Heterogeneity*** in dataspaces, the annotation can be improved and refining of schema mappings will become easy. It will also reduce the query processing time on user's query by removing the Naming Conflicts, Conflicts between entities, Conflicts between Attributes, wrong data conflicts, incomplete data conflicts, Noisy Data conflicts etc. in Dataspaces. The main problem arising here is to determine that order in which the candidate matches should be confirmed by the user on the basis of their feedback, which is proved to be most beneficial to a dataspace.

In this paper [2], a new framework i.e. decision-theoretic framework was developed for the correct ordering of candidate matches being produced and after that those will be confirmed by the user. In this framework, a key concept the value of perfect information (VPI) is used. With the help of VPI, a true value for some unknown can be determined and also the information related to the candidate matches i.e. whether it is correct or not can be obtained. The various steps of this algorithm are as follows:

1. Develop a method for ordering candidate matches using decision-theoretic framework.
2. Use the utility function at any given state of a dataspace.
3. Make a decision by comparing the utility of dataspaces before and after the user confirmation of a candidate match.
4. Based on the query result (obtained in step 3), the utility function is devised (by the use of Thresholding).
5. Finally, the set of approximations are developed to estimate the utility of dataspaces in an efficient manner.

The *advantage* of this algorithm is that the entire candidate matches produced from different or multiple mechanisms always be considered in a uniform fashion.

2.2 Feedback-Based Annotation, Selection and Refinement of Schema Mappings for Dataspaces [3]

In dataspaces, as there are different schemas defined, so there is a need to map these schemas and also to annotate, select and refine different schema mappings. For this purpose, K. Belhajjame et.al [3] has proposed different algorithms in this regard. These algorithms based on the approach in which the schema mappings were annotated iteratively with estimates of precision and recall on the basis of user's feedback [4]. Low precision and recall indicates that the quality of schema mappings is poor. User feedback can also be used to improve the quality of existing mappings through refinement. The paper contributes in annotating schema mappings (having an estimate of precision and recall), schema mapping selection (in terms of precision and recall), mapping refinement (include mutation and cross-over algorithms) and an evaluation showing effectiveness of all of these. A data integration system can be defined as a combination of different schemas, data sets, integration schema on which queries are fired and schema mappings. The mapping annotations like True Positive, False Positive, and False Negative can also be used here for the user feedback.

The first concept described in this paper is ANNOTATING SCHEMA MAPPINGS. Annotation simply means to indicate a schema mapping as correct or incorrect. Correct in the sense that it retrieves all the results as user expectations. Annotations may be Cardinal or Ordinal.

- Cardinal Annotations: - To get a quantified result of the quality of mappings in an information schema by calculation precision and recall using the user feedback in the form of mapping annotations as described above.
- Ordinal Annotations: - In this scheme, the dependencies are made between the candidate mappings in terms of the tuples, received from user feedback. This one is very much time consuming because for every two candidate mappings, the results of the source queries needs to be compared.

SELECTING SCHEMA MAPPINGS: To select from a set of candidate mappings, the mapping annotations are needed. The mapping selection can be considered as an optimization problem because all the users do not have same requirements in terms of precision and recall. In the mapping selection technique, a selection method is generated in which the recall obtained from the results need to be maximized where as the precision needs to be higher than a threshold value (given by the user).

REFINING SCHEMA MAPPINGS: Refinement is a process by which the quality of candidate mappings can be improved. In this process, by the help of user feedback, new mappings can be constructed from the existing ones. There may be two kinds of refinement:-

- ***Refining Mappings to Reduce the Number of False Positives:*** To refine a candidate mapping, the source queries are modified in such a way that the number of false positives get reduced. For this purpose, four operators of the relational algebra can be used: Join, Selection, intersection and Difference. The number of false positives returned may be reduced by applying these operators along with the source query and the other source relations or schemas according to the type of operator.
- ***Refining Mappings to Increase the Number of True Positives:*** In this refining method, the operator Union is used in which this operator is apply between the source query and any other query which retrieve true positives that are not returned by that mapping.

There is a question arise in refining schema mappings that how to explore all the potential mappings so that the best possible mapping can be discovered because after refinement, the quantity of mapping obtained is very large since the different operators can be recursively applied to the schemas.

To the solution of this, the cross-over and mutation operators are used. Using cross over operators, new mappings can be constructed by combining good parts of existing techniques and mutation operators give an optimal solution by diversifying all the candidate mappings [10]. There are various algorithms discussed for mutating a candidate mapping (refine mappings and mutate mappings algorithms) and combining candidate mappings (include cross over mapping algorithm).

2.3 Building Data Integration Systems: A Mass Collaboration Approach [5]

To build an effective data integration system, an approach i.e. mass collaboration approach is proposed in this paper by A.Doan et.al. A data integration system is a system in which data can be retrieved from various sources. The results of any query are gathered from different source schemas and then combined at one place so that the best answer can be searched from there. The high cost is a big problem in making a data integration system. In this paper, the problem of cost in making an integration system can be solved by dividing it 'thinly' among producers and consumers. The data integration systems include a large amount of work done by hand and other error prone processes. This is also a disadvantage in making these systems. This problem also solved in this paper. The idea behind the Mass Collaboration approach is to set the values of all finite parameters of a data integration system initially. The steps of MOBS (Mass Collaboration to Build Systems) approach are:-

1. Build the source schemas and mediated schema (where all the results of a query get combined).
2. The system parameters are defined from the semantic mappings of elements of mediated schema.
3. Assign initial values to the system parameters.
4. The system shell is constructed then and deployed on the Internet.
5. Ask for user feedback.
6. The values of system parameters are readjusted according to user feedback, until these values converge.

In step 2, the semantic mappings are treated as the system parameters. In fact, the other features like the source schemas etc. can also be considered for this.

The advantage of this approach is that it reduces the cost of build a data integration system but along with this, it has many challenges. Some challenges are:-

- **System parameters-** What to be taken to be the system parameters? For system parameters, we can take semantic mappings, source schemas etc. The parameters should be application-specific.
- **Setting values of parameters initially-** Any random value can be given to the parameters or may be initialized by using any tool. If these values are closer to the correct values, then the system will converge soon.
- **Starting with a partially correct system-** An initial incorrect system will be obtained if all system parameters were set as described above, which cannot be used by users because it also gives incorrect results. Due to this, the starting should be from a correct subsystem so that the users get their approximate results.
- **Enticing users for feedback-** To convince users for feedback is a difficult problem. Some ideas for this purpose may include:
  Forced feedback- When user asks any query, automatically a feedback question comes on the screen.
  Feedback with instant gratification- If the user needs more details about his query other than the result then he has to give his feedback.
  Feedback with delayed gratification- The user willingly provide feedback if they know that this feedback will bring long-term benefits to them.
- **Types of Questions to be asked from users-** The questions should be simple so that the user will easily provide his feedback.

The other challenges may be how to handle ignorant users, how to combine user feedback at one place, how to increase the quantity of feedback etc.

2.4 Pay-As-You-Go Mapping Selection in Dataspaces [6]

In this paper, the details of DSToolkit are explained i.e. how to work in a DSToolkit. DSToolkit is a dataspace management platform or a system in which users can easily provide their feedback on results of queries over an integration schema. It is used for initializing and improving dataspaces on the basis of user feedback. After user feedback, the mappings are annotated using respective precision and recall.

To create mappings between integration schema and source schemas is a complicated and time-consuming task. Due to this, the dataspaces are developed which works in a pay-as-you-go fashion.

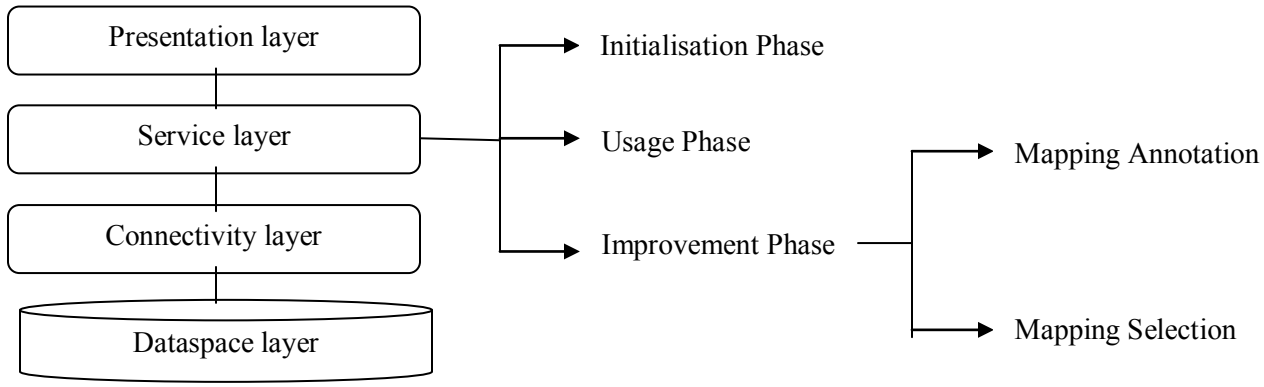
The features of DSToolkit are:-

(i) It is a dataspace architecture in which the functions like bootstrapping, improvement and maintenance are supported.
(ii) The schema mappings are annotated incrementally based on user feedback on query results.
(iii) Selection of mappings at query execution time which is based on user-specified QoS criteria.

The DSToolkit works on a layered architecture.

.

Figure1. DSToolkit Architecture



The various layers are-

- Dataspace layer
- Connectivity layer
- Service layer(initialisation, usage and improvement)
- Presentation layer

Initialisation phase- Also known as Bootstrapping. In this phase, the data resources are identified and then integrated.
Usage phase- In this phase, the queries are fired from the integrated data sources.
Improvement phase- This phase includes Mapping annotation and Mapping Selection.

Mapping Annotation :-> The user feedback on result tuples can be taken in the form of:

- True Positive (TP): Whether the tuple was expected to be present.
- False Positive (FP): Whether the tuple was expected not to be present.
- False Negative (FN): A tuple that was not returned but users expected to see it. It can be given as an input.

Using these values, for a mapping m (relative to user feedback UF), the precision and recall are calculated using the formulas:

$$\text{Precision}(m, UF) = \frac{|TP(m, UF)|}{|TP(m, UF)| + |FP(m, UF)|}$$

$$\text{Recall}(m, UF) = \frac{|TP(m, UF)|}{|TP(m, UF)| + |FN(m, UF)|}$$

Mapping Selection: After mapping annotation, the mappings are selected from that information so that the query posed by a user can be answered. It is the choice of the user that which precision or recall they want to choose. In mapping selection, the user can evaluate the same query or different queries by rerunning them with different precision or recall targets.

2.5 User Feedback as a First Class Citizen in Information Integration Systems [7]

Since, user feedback plays an important role in improving the quality of already existing systems in Dataspaces. So, to achieve the maximum benefits from user feedback, it is considered as a first class citizen. There are several advantages of this which are shown in this paper. The idea behind this paper is that the existing proposals has some assumptions like a data integration system either has a single user, or all the users may be have same requirements, or like user requirements do not change over time etc. These assumptions may not be true every time. There are conflicts between the feedback supplies by the user because two users may also have different requirements and also the requirements of a user may change over time. This phenomenon is known as the feedback inconsistency. Various examples are given in this paper in support of this concept.
There are certain predicates on which the validity of feedback can be checked:

- exists(obj) = true ; if object obj exists, false; otherwise.
- Valid(uf) = true ; if feedback instance uf is valid, false; otherwise.
- Invalid(uf) = true ; if feedback instance

uf is invalid,
false; otherwise.

- hasUnknownStatus(uf) = true ; if status of uf is unknown, false; if uf is either valid or invalid.

The values of these predicates can be change over time. Thus, there are more two predicates which are specific to time: holdsAt and happens. On the basis of feedback provided by users, the users can be clustered. The groups of users are formed according to their requirements. The advantages of this are:-

- Within a cluster, the collective feedback can be used which improve the quality of the integration system.
- Multiple information integration systems need to be created to handle the users having different requirements.
- Clustering helps in grouping the users (having same requirements) of different integration systems into one group.
- The remaining users can also be identified easily.

There is also the use of a technique named "Collaborative Filtering", that is used for learning user feedback and what user prefers more because it is not possible to take feedback from a user who just join the data integration system. To learn the feedback, precision and recall can also be used.

2.6 Soliciting User Feedback in a Dataspace System [8]

In the pay-as-you-go approach, after making candidate matches, it is necessary to incrementally confirmed them through user feedback. There should be a ordering of these matches or we can say that in what order the matches should be confirmed. This is the main challenge here. The idea behind this approach is to determine the ordering in which the user feedback can be solicited for confirming candidate matches. In soliciting user feedback, the focus is on the output of a single mechanism. The goal is to reduce the uncertainty from the obtained matches regardless of the matter that how important these matches are to the queries in dataspaces. The work done in this paper is very much similar to that of paper [2]. The difference is that in soliciting user feedback, only output of a single mechanism is taken into account whereas in the latter one, multiple mechanisms have been focused.
The decision-theoretic framework is used here including the concept of value of information (VPI). This also explains the design of Roomba, a system that incorporates the decision-theoretic framework to guide a dataspace in soliciting user feedback in a pay-as-you-go manner. There are certain strategies for ordering the matches. There is a big scope of extending this work in future.

2.7 Incrementally improving dataspaces based on user feedback [9]

Since, the process of schema mapping is very much time and resource consuming. To resolve this problem, the schema mappings can be derived from information obtained using schema matching techniques. In this paper, the algorithms for annotating, selecting and refining schema mappings are given on the basis of user feedback as in [3]. Annotation of mappings can be done using precision and recall after taking user feedback. It has been also evaluated that the more user feedback is there, the better is the quality of the estimates. This mapping annotation process is more cost-effective and also may lead to high error rate due to feedback inconsistencies. After that, a method for selecting the best mapping is given. At last, by the method of refinement, the mappings having better quality can be constructed from initial candidate mappings. In the process of annotating queries over integration schema using user feedback, the result generated is that the lower the amount of query selectivity, the smaller the error in the precision and recall estimated. The strategies for mapping can be incorporated within the DSToolkit, which is explained above.

III. COMPARISON

This section presents a theoretical comparison among the various approaches presented in the previous section.

Table 1: Comparison Table

APPROACH	FEEDBACK TYPE	FEEDBACK PROCESSING	OBJECTS ON WHICH FEEDBACK IS GIVEN	ADVANTAGES	DISADVANTAGES
Pay-as-you-go User Feedback	Confirmed matches in the form of *true* and disconfirmed as *false*.	By the use of given feedback, calculate Value of Perfect Information (VPI).	Dataspace D, set of candidate matches $M=\{m1,\ldots,m_l\}$	Quality of the results obtained gets improved using this feedback.	Highest Expected Utility is not achieved because of using the thresholding concept only for confirmed matches.
Feedback - Based Annotation, Selection and Refinement of Schema Mappings	True Positive (TP), False Positive (FP) and False Negative (FN).	Annotate schema mappings, calculating Precision, Recall and F-measure (relative to user feedback that combines both precision and recall and use it for ranking candidate mappings). Schema mappings are selected and then refined in order to obtain the best results.	User Feedback, Precision, Recall and a set of candidate mappings.	As more feedback is provided by the user, better quality mappings are derived through refinement.	Mapping Selection is an optimization problem here and also inconsistencies may exist due to change in user expectations.
Pay-As-You-Go Mapping Selection	Users provide their feedback in the form of comments on results of queries posed over an integration schema in the form of TP, FP and FN.	First, mappings are annotated and then on the basis of user feedback, the system selects the mappings in order to meet user requirements.	An integration schema and some mappings.	There is an improvement in query results because it does not need any expert knowledge from the user which results in improving the Dataspaces.	In order to achieve better results by mapping selection, the recall is maximized over the union of the results returned, which degrade the processing of the algorithm.
User Feedback as a First Class Citizen in Information Integration Systems	Feedback is in the form of a tuple i.e. (obj, t, u, k). User u annotates the object obj using term t, k is the value of different kinds of feedback provided by the user, k= {TP, FP, FN}.	Once the feedback is received, the feedback validity needs to be finding out. Then, Clustering of users is done on the basis of feedback. At last, the feedbacks are overlapped.	A result tuple, an attribute along with its value and a candidate query.	It removes the problem of feedback inconsistency.	The smaller the overlap between users in terms of feedback, the poorer is the quality of clustering and overlapping is totally based on the feedback provided by different users.

IV. CONCLUSION

In this paper, a detailed description of the techniques used for taking feedback from user in the context of dataspaces is given. The user feedback can be used for annotating, selecting and refining schema mappings. A brief overview of all the techniques for this purpose is presented in this paper along with the advantages and disadvantages of the existing algorithms. The pay-as-you-go approach for user feedback improves the quality of results obtained and takes feedback in the form of true or false. The work done in this approach is by manually. Then, there is an approach in which the feedback is taken in the form of TP, FP and FN, i.e. whether the user Expected (TP) or Not Expected (FP) the results. Similarly, there are various techniques shown in this survey. From these techniques, it can be concluded that proper user feedback is very important in order to improve the dataspace systems.

REFERENCES

[1]. Franklin, M. & Halevy, A. and Maier, D., "From databases to dataspaces: a new abstraction for information management," *ACM Sigmod Record*, 2005.

[2]. S. R. Jeffery, M. J. Franklin and A. Y. Halevy, "Pay-as-you-go User Feedback for Dataspace Systems," in *SIGMOD'08*, 2008.

[3]. K. Belhajjame, et al., "Feedback-Based Annotation, Selection and Refinement of Schema Mappings for Dataspaces," in *EDBT* 2010.

[4]. C. J. van Rijsbergen, *Information Retrieval*. Butterworth, 1979.

[5]. A. Doan and R. McCann, "Building Data Integration Systems: A Mass Collaboration Approach".

[6]. C. Hedeler, et al., "Pay-As-You-Go Mapping Selection in Dataspaces," in *SIGMOD'11*, 2011.

[7]. K. Belhajjame, et al., "User Feedback as a First Class Citizen in Information Integration Systems," in *CIDR*, 2011.

[8]. S. Jeffery, M. Franklin and A. Halevy, "Soliciting User Feedback in a Dataspace System".

[9]. K. Belhajjame, et al., "Incrementally improving dataspaces based on user feedback," *Information Systems (2013).*

[10]. C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv*., 35(3):268–308, 2003.

[11]. Podolecheva, M., Prof, T., Scholl, M. & Holupirek, E., "Principles of Dataspaces," Seminar From Databases to Dataspaces Summer Term 2007 Citeseer, 2008.

[12]. Mirza, H. T., Chen, L. and Chen, G., "Practicability of Dataspace Systems," JDCTA*: International Journal of Digital Content Technology and its Applications*, 4(3), 233–243 (2010).

