International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 02– Issue 06, November 2013

A Survey of the State of Dataspaces

Shibwabo Bernard Kasamani [1], Ateya Ismail Lukandu[2]
[1,2]Faculty of Information Technology, Strathmore University
Nairobi- Kenya
[1]Email: bshibwabo {at} strathmore.edu

Wanyembi Gregory Wabuke [3]
[3]Department of Computer Science, Masinde Muliro University of Science and Technology
Kamamega- Kenya

Abstract— **This paper presents a survey of the state of dataspaces. With dataspaces becoming the modern technique of systems integration, the achievement of complete dataspace development is a critical issue. This has led to the design and implementation of dataspace systems using various approaches. Dataspaces are data integration approaches that target for data coexistence in the spatial domain. Unlike traditional data integration techniques, they do not require up front semantic integration of data. In this paper, we outline and compare the properties and implementations of dataspaces including the approaches of optimizing dataspace development. We finally present actual dataspace development recommendations to provide a global overview of this significant research topic.**

Keywords— **Dataspaces, Systems integration, Spatial databases, Entity collaboration, Geo data, Data management.**

I. INTRODUCTION

As the volumes of data storage increases within and across enterprises, there is a growing need to develop efficient and effective techniques of data management. With the increase in the amount of structured, semi-structured and unstructured data available on the web as well as local data stores, the impact has been that new opportunities for using data integration technologies have been created. However, in spite of the long standing research work in data integration, this technology seems to have had a limited impact in practice. To a large extent, data integration mechanisms are manually-coded and tightly bound to specific applications. The limited adoption of data integration technology is partly due to its cost-ineffectiveness [1].

The problem of data integration has been investigated for a relatively long period of time spanning about two decades with the aim of providing end users with a transparent access to data sets that reside in multiple data sources and are stored using heterogeneous representations. Data integration has numerous potential applications, e.g., it can be used for providing cross-querying of data stored in databases that belong to multiple departments or organizations, or to enhance collaboration in large scientific projects by providing investigators with a means for querying and combining results produced by multiple research labs [1].

More precisely, the specification of schema mappings (in such a way that, data structured under the source schemas is transformed into a form that is compatible with the integration schema against which user queries are issued) has been found to be both time and resource consuming, and has also been determined as a critical bottleneck to the large scale deployment of data integration systems [2].

Dataspaces is a current technique of managing data. Since its envision in 2005, dataspaces has growingly been fronted as the new technique of data integration [3]. Data integration is an important research topic since it aims at providing transparent access to data that is stored in various data repositories that are often using different underlying data models. Various attempts have been proposed towards the development of dataspaces by developing dataspace support platforms (DSSP). However, the development of a complete global dataspace is still an unaccomplished research concern.

Dataspaces are considered to be unique and special since they eliminate the requirement for up front semantic data mapping as is the case for traditional data integration approaches. Dataspaces are further described as not really a data integration approach but as a data coexistence approach [3], [4]. This way data integration can be provided on a pay-as-you-go fashion also described as on demand or incrementally.

Efficiently evaluating dataspace developments so far is crucial for the determination of the best way forward. The initial suggestions about the requirements of a dataspace outline important characteristics of a dataspace [5] followed by subsequent implementation attempts have made dataspaces an interestingly growing technique of systems integration. The development of dataspaces requires the development of practical algorithms, flexible design as well as further successful implementation of the same so as to address the concern by [3] that most integration approaches lack the speed, flexibility and economy (integration on-demand) that many organizations need today in a data integration solution.

Since late 2005, a relatively tremendous amount of research has been based on focusing on, or relating to dataspaces. However, few real implementations exist that fulfil the all the principles or requirements outlined by [2]. Ref. [5] provides earlier work that deals with the architecture and functionality of a Dataspace Management System. They further provide a more detailed definition of a dataspace as a set of software programs that controls the organization, storage and retrieval of data in a Dataspace. They had that the dataspace also handles the security and integrity of the Dataspace. A recent survey on data integration is conducted by [4], though it focuses broadly on data integration approaches as opposed to dataspaces. Fig. 1 shows the trends in data integration.

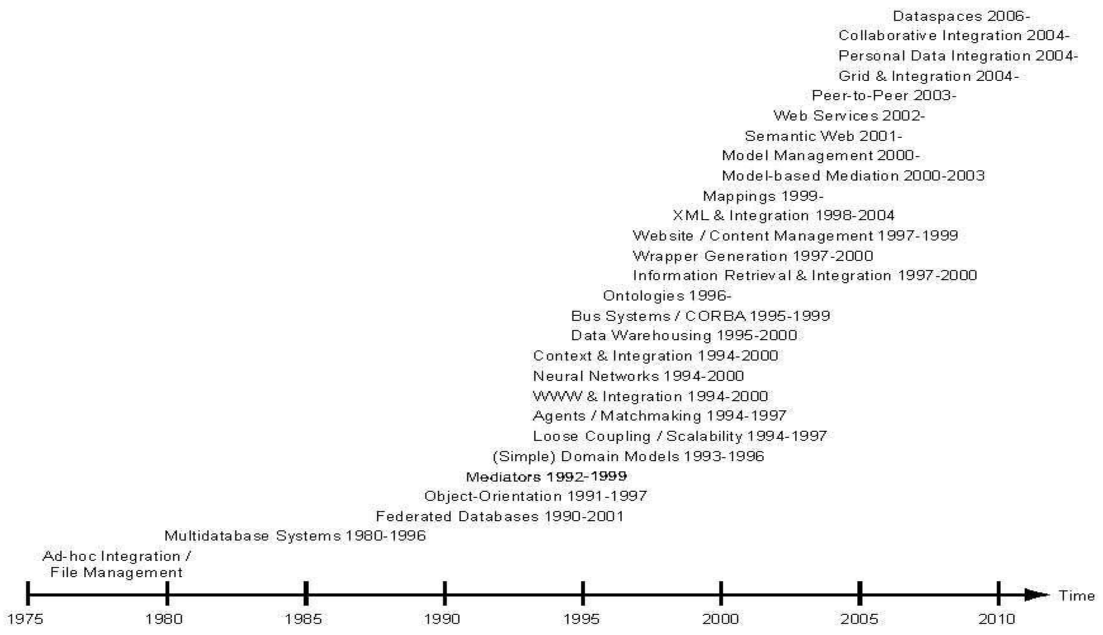

Figure. 1. Data Integration Research Trends over Time (Adopted from [4]).

The aim of this paper is thus to provide a global and in depth overview of more than 7 years of research about dataspaces and closely related concerns. We begin this by defining dataspaces and related concepts in Section II. Then, we present and further discuss the various dataspace design and implementation attempts in Section III. We conclude this paper and provide insight on the way forward regarding dataspace related research in section IV.

II. PRELIMINARIES

In this section, we first formally define all the concepts used in this paper (Section A). We then establish some existing examples that illustrate throughout the paper how dataspaces generally operate (Section B).

A. *Definitions*

1. *Data Integration*
   Data Integration is a concept in information management that aims at providing transparent access to existing data. As individuals and organizations increasingly store more and more data especially in divergent data stores, the need to integrate data also increases.
2. *Dataspace*
   A dataspace is known as a data integration approach that provides for data coexistence in a space of data incrementally. This way, the dataspace will consist of an infrastructure that is customizable by users to define their domain of interest which forms the domain of a dataspace.
   Let D be a set of all data in a dataspace and Q (D) as a query on the dataspace. Then a query result denoted as R is in such a way that it can be expressed by the expression as $R \subseteq D$ which describes that every element in the query result R is a member of the dataspace D.
3. *Dataspace Support Platform*
   A dataspace Support Platform (DSSP) is an infrastructure that supports a dataspace. In order to have a working dataspace, it is critical to implement relevant techniques, policies and algorithms towards the implementation, which together define the DSSP.
4. *Dataspace Composition*
   A dataspace is described to consist of entities and their relationships. The process of developing a Dataspace Support Platform is to find ways of better representing these entities as well as the relationships between them.

B. *Why Dataspaces?*

Dataspaces are clearly distinguished from traditional data integration approaches due to the fact that they provide for integration on a pay-as-you-go fashion. This way it is cheaper. More importantly, dataspaces do not require upfront effort for semantic integration; they focus on data co-existence instead. Additionally, dataspaces provide higher degrees of scalability due to the perceived nature of the entity relationships [1], [2], [3], [4].

C. *Examples*

One example of a dataspace described as a personal dataspace is described by [2] as a personal dataspace in which users access data stored in a set of personal data repositories.

That set of repositories may include private file systems that currently exist on a user’s desktop as well as private e-mails of a particular user. In the case of personal dataspaces just like the case of other dataspaces, users often experience difficulties understanding which items spread across their repositories/sources are related to each other in the same context. Although users are likely to search their data sources with search engines, the results obtained by these systems are not enriched with contextual information. Users may typically desire to access the various versions of a certain file that exist in their dataspace, view files as well as emails worked on approximately the same time, or extract emails in the same project of a certain document.

A second example of a dataspace which is available as a public dataspace is called Google Base. It is further described by [6] as a very large, self-describing, semi-structured, heterogeneous database. Google Base certainly consists of a set of tuples with attribute values. Each tuple entry T_e is regarded to consist of a number of attributes with matching values. An illustration of a dataspace tuple is shown in Fig. 2.

T_e: { (name : Galaxy), (color : white), (manu : Samsung), (tel : 7899), (addr : 456 Talk to us, Seoul) , (website : samsung.com)};

Figure 2. An Illustration of a Dataspace Tuple.

In this case, the tuple in the dataspace set can be considered as a tuple in an existing dataspace. In this second example, the data set is enormously sparse due to the heterogeneity of data, which are continuously contributed by users around the world.

III. ANALYSIS AND DISCUSSION OF DATASPACE CONSTRUCTS

We review in this section the various dataspace implementations found in the literature. We then later conclude this section by discussing their features in Section 3.3.

A. Dataspace Principles

Dataspaces have been proposed as a data management abstraction for diverse interrelated applications [3]. Dataspace Support Platforms (DSSPs) are systems that should be built to provide the required services over dataspaces. Unlike data integration systems, DSSPs do not require full semantic integration of the sources in order to provide useful services. There is need for query answering in DSSPs, the DSSP's ability to introspect on its content, and the use of human attention for the purpose of enhancing the semantic relationships in a dataspace [3].

Traditional data integration systems require semantic integration before any services can be provided. Hence, although there is not a single schema to which all the data conforms and the data resides in a multitude of host systems, the data integration system knows the precise relationships between the terms used in each schema. As a result, significant upfront effort is required in order to set up a data integration system [3].

The following properties distinguish DSSPs from traditional databases as described by [3]:

i. A DSSP must deal with data and applications in a wide variety of formats accessible through many systems with different interfaces. A DSSP is required to support all the data in the dataspace rather than leaving some out, as with DBMSs.

ii. Although a DSSP offers an integrated means of searching, querying, updating, and administering the dataspace, often the same data may also be accessible and modifiable through an interface native to the system hosting the data. Thus, unlike a DBMS, a DSSP is not in full control of its data.

iii. Queries to a DSSP may offer varying levels of service, and in some cases may return best-effort or approximate answers. For example, when individual data sources are unavailable, a DSSP may be capable of producing the best results it can, using the data accessible to it at the time of the query.

iv. A DSSP must offer the tools and pathways to create tighter integration of data in the space as necessary.

The participants in a dataspace are the individual data sources: they can be relational databases, XML repositories, text databases, web services and software packages. They can be stored or streamed (managed locally by data stream systems), or even sensor deployments. A dataspace should be able to model any kind of relationship between two (or more) participants [3].

Dataspaces can be nested within each other (e.g., the dataspace of the CS department is nested within the dataspace of the university), and they may overlap (e.g., the dataspace of the CS department may share some participants with the EE department). Hence, a dataspace must include access rules between disparate dataspaces. In general, there will be cases where the boundaries of a dataspace may be fluid, but we expect that in most of the cases the boundaries will be natural to define [3].

B. Schema Mappings and User Feedback

An important aspect of dataspace development is the development of candidate mappings as well as a description of the model for defining user feedback. It has been found that a data integration system is essentially composed of four elements. These elements are described as: the schemas of the data sources, the data sets to be integrated, an integration schema over which users pose queries, and schema mappings that specify how data structured under the schemas of the sources can be transformed and combined into data structured according to the integration schema [7].

A schema mapping can be defined by the pair (q_i, q_s), whereby q_i and q_s are any two queries of the same arity over the integration schema and the source schemas, respectively. This mapping ideally specifies that the concepts represented by the queries q_i and q_s are semantically equivalent [8]. For the purposes of dataspace schema mappings, it is possible to restrict ourselves to mappings that relate one element in the integration schema to a query over the source schemas: these mappings referred to as global-as-view mappings [8]. It is also possible to adopt the relational model for expressing integration and source schemas. A schema mapping m can be defined by the pair m = (r_i ; q_s), where r_i denotes a relation in the integration schema, and qs denotes a relational query over the source schemas [1]. This way, it is possible to use *m.integration* to refer to r_i, and *m.source* to refer to q_s.

Existing schema matching techniques can be used to produce the input for algorithms capable of automatically generating the mappings between the integration schema and the source schemas (e.g., [9]). Multiple matching techniques can be applied, each of which could result to multiple mapping candidates for populating the elements of the integration schema. In order to answer a user query u_q, which is executed against the integration schema, each relation r_i participating in u_q requires to be reformulated in terms of the source relations using a mapping candidate. This raises the question as to which mappings among the candidate mappings of r_i to use for answering a user query [1].

It is possible to label candidate mappings by scores that are obtained from the confidence of the matches used as input for the generation of mappings (e.g., [10]). This indicates that the candidate mappings that have the highest scores can be used for reformulating users' queries. However, due to the fact that the confidences of matches, and henceforth the scores of mappings, are generated based on heuristics, there is no guarantee that the mapping with the highest score reflects the exact expectations of dataspace users [11, 12]. In fact, in a data integration environment, it is rare that the content of the integration schema is available, and therefore instance-based matchers may not be applicable to match the source schemas to the integration schema. We can conclude that, the probability that the scores associated with the mappings lack accuracy can be higher than in situations in which the contents of the schemas to be matched are available, e.g., in data exchange [7].

Although it is possible to automatically derive schema mappings schema mappings using existing mapping generation techniques [13, 14], the outputs of the mappings by these techniques may not necessarily match the expectations of users. There has been researcher in an effort to address the issue of mapping verification within the context of data exchange. Various authors [15] presented a debugger for understanding and exploring schema mappings. For this purpose, they provide computations, and display on request, the relationships, termed routes, between source and target data with the schema mapping in question. Other authors proposed Spicy [11], as a system for verifying the quality of mappings between a source and target schema. In order to confirm a set of schema mappings, their source queries are executed against the source schema and the results extracted are compared with instances from the target schema, the contents of which are assumed to be available. The results of this comparison are meant to identify incorrect mappings, and to suggest to designers the mappings that are likely to be accurate.

By using the tools indicated earlier, the verification of schema mappings takes place prior to setting up the data integration system, potentially incurring a considerable up-front cost [2, 3]. This is contrary to the dataspaces vision since it advocates for a scenario where the annotations and refinement of the candidate mappings should be accomplished as the data integration proceeds incrementally.

A more recent research [1] explores a different approach in which generated schema mappings co-exist, and are verified in a pay-as-you-go fashion. They consider a scenario whereby the data integration infrastructure is setup using input schema mappings that are obtained using mapping generation techniques. These mappings are then incrementally annotated with estimates of precision and recall [16] derived on the basis of feedback from end users. This way, users are not provided with a set of (probably complex) mapping expressions; rather, they are provided with a set of answers to a query executed against the integration schema and which was answered using one or more candidate mappings. The user further examines and comments on the returned results using the following sort of feedback:

- That a certain tuple was expected in the answer.
- That a given tuple was not expected in the answer.
- That an expected tuple was not retrieved.

The types of feedback described by [1] are tuple-based since they comment on the correctness of the membership relation between tuples and the set of result obtained by a set of mappings. It is possible to refine feedback even further, especially; a user can indicate that a given attribute of r_i cannot have a certain value. As in information retrieval [23], it is assumed that users supply feedback voluntarily: it is not mandatory for them to comment on every single result they are presented, rather, they provide feedback on the results of their choice.

In order to realize the types of feedback introduced earlier, it is possible to define a feedback instance *uf* supplied by the user by the tuple [1]. This is presented in (1):

$$uf = \{AttV, r, exists, provenance\} \quad (1)$$

where r is an existing relation in the integration schema, *AttV* is a set of attribute-value pairs $\{att_i ; v_i \}i; 1 \leq i \leq n$, such that att_1, ... ,att_n are attributes of r, and v_1, ... ,v_n are their respective values. *exists* is a boolean specifying whether the attribute value pairs in AttV conform to the user's expectations.

It is not true to say that all users of information integration systems will posses equal requirements in terms of precision and recall [1]. As an example, consider a data integration system that provides access to existing proteomic data repositories. A drug designer who executes queries to such a data integration system may need high precision; the existence of false positives in query results may result to the further costly investigation of inappropriate candidate drugs. Conversely, an immunologist utilising a proteomic data integration system for purposes of identifying the proteins responsible for an infection may accept low precision, since further investigation is likely to result to the discovery of new proteins associated with the infection under investigation [1].

With regards to Schema mappings and User feedback, there are further problems that are yet to be dealt with [1] in order to realise the dataspace vision. Some of the problems include inconsistencies that may exist in user feedback, which may transpire due to factors like changes in user expectations. There is need to analyse the impact that such inconsistencies can have on mapping annotations.

C. iDM: As a Unified and Versatile Data Model for Personal Dataspace Management

Personal information consists of extremely heterogeneous data combination of emails, XML, and word documents, images, audio files, address book entries, and so on. Personal information is typically stored in files scattered among various file systems, multiple machines (local desktop, network share, mail server), and even important, different file formats (XML, LATEX, Office, email formats, etc.). The first attempt to represent all of the highlighted data in a single and simpler data model is presented by [18] through the iMeMex Data Model (iDM) for personal information management. They indicate that the approach provides the following advantages:

(1) iDM clearly differentiates between the logical data model and its physical representation,

(2) iDM is powerful enough to represent XML, relations, files & folders and cyclic graphs in a single data model,

(3) iDM is able to represent the structural contents inside files as part of the same data model,

(4) iDM is powerful enough to represent extensional data (base facts), intensional data (e.g. ActiveXML), as well as infinite data (content and data streams),

(5) iDM enables a new class of queries that are not available with state-of-the-art PIM tools.

In as far as the authors [18] define a Model for managing dataspaces, the resulting solution lacks in terms of fulfilling the dataspace principles earlier defined. They highlight some issues relevant to personal dataspace management that are orthogonal to the proposed model, though easier to address once a data model similar to iDM is in place:

1. Versioning. A PDSMS keeps track of all changes made to the dataspace. Like classical versioning techniques, logically, each change creates a new version of the whole dataspace. With iDM, the implementation of versioning is simplified due to the representation of the entire dataspace of a user in one model.

2. Lineage. Data lineage is keeping the history of all data transformations that originated a given resource view. For example, when a user copies a file into another and then modifies the new file, the system should keep for the new resource view the information about its provenance. With a unified model such as iDM, it is possible to keep lineage information across data sources and formats.

In addition to the issues highlighted, the iDM model has not been adopted widely due to several reasons. The overall goal of dataspaces is to have a plug and play platform for systems integration. iDM mainly provides integration for Personal information. More features that need further improvements or development include Integration of updates from data sources, user feedback, Cost-based query optimization, Scalability (Support for larger datasets > 25 GB, scaling beyond 1 TB using distributed instances) as well as the incorporation of machine learning techniques.

D. Constructing a Dataspace Based on Metadata and Ontology for Complicated Scientific Data Management

Some work [19] has been done to introduce a framework of constructing a dataspace based on metadata and ontology, particularly for complicated scientific data management. The solution is initially supported by the ontology. Ontology refers to an explicit specification of a conceptualization of the knowledge in a certain domain or even on a larger scale [20], and it provides machine understandable definitions for concepts and relationships between these concepts. Ontology is key in interoperation and content communication and has been adopted by fields such as Semantics Web, information management, and digital libraries.

The proposed method sets up a metamodel independent from the various platforms of data sources, and then applies the metamodel to describe the participants of the dataspace and the relationship between them, that is producing metadata to describe the dataspace. Based on the unified metadata, the search and query service can further be provided for all the data within the dataspace [19].

A framework is described by [19] for Complicated Scientific Data Management. This is presented in Fig. 3.

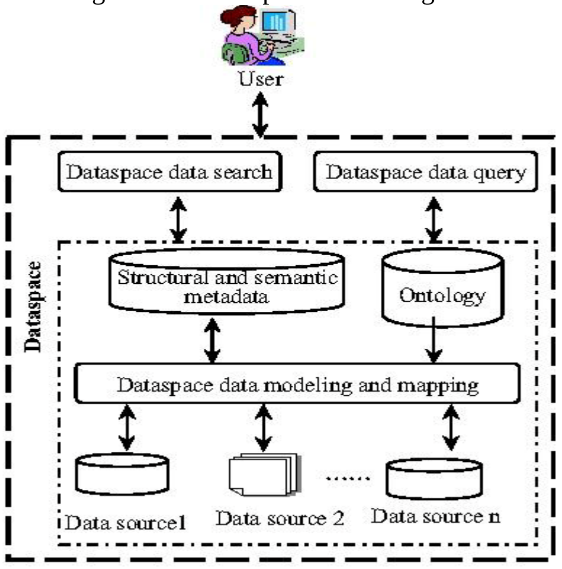

Figure 3. The framework for Complicated Scientific Data Management (Adopted from [19]).

The proposed approach sheds more focus on theoretical aspects but lacks to provide important working implementation of the dataspace based on the proposed framework. They further propose the need for future work to combine the technologies of metadata and ontology and looking for a new technological path to implement the whole framework. No testing has been suggested on the provided model as well.

E. Dataspace Realization on the Grid

Ref. [21] describe that, no effort has been devoted to realization of Dataspace concepts on the Grid. The paper proceeds to propose the architecture of a Dataspace Management System and further discuss how some current components of the Grid technology can support the future implementations of such an architecture.

The authors [21] discuss the requirements of applications on a Dataspace by the means of the Dataspace Environmet components depicted in Fig. 4.

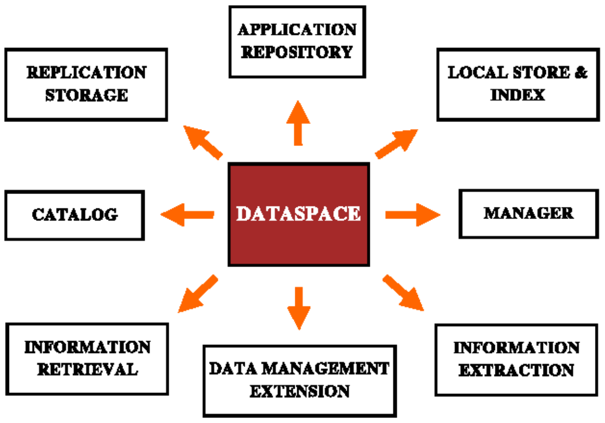

Figure 4. The Dataspace Environment (Adopted from [21]).

1. Dataspace Management Workflows

In this Section most relevant workflows for managing Dataspaces are described. First of all, a Dataspace is defined and labeled. The next step is to add participants, by registering the new data sources. Therefore the user enters a reference to the participant. Then data descriptions can optionally be entered, either by hand or by entering a reference to a description file if available. Containing these information the participant will be registered within the Catalog and the Metadata Repository. All these steps, including the definition of relationships between two or more participants can be integrated into the processing step “Add Participant” within a workflow. Search and Query, Information Extraction and Data Replication are other possible workflow components that can be defined within a Dataspace ManagementWorkflow. Depending on domain specific applications, more particular workflows can be defined.

2. Grid Technology Support for Dataspaces

Grid computing has been identified as an important new technology by a remarkable thread of scientific and engineering fields as well as by many commercial and industrial enterprises [22]. Its goal is to share and manage geographically distributed computer resources and data across enterprises, industry or workgroups independently of the operating characteristics of their computer systems. It can be used to temporarily increase computational power and storage needs on demand. So far, essentially all major Grid projects have been built on protocols and services of the Globus Toolkit [23], which is an open source software toolkit.

The application of Grid tools has been proposed to solve the dataspace development challenge [21]. The tools include GridFTP, Replica Management, The Metadata Catalog Service (MCS), Storage Resource Broker (SRB) and OGSA-DQP. However, the discussion is presented in a theoretical sense with the expectation that they will be implemented in future.

F. Modelling Dataspace Entities and their Associations

More recent work has been conducted relating to the representation of entities participating in a datspace as well as their relationships by Shibwabo, Wanyembi and Ateya [24]. The authors model a dataspace using the set theorem with entity mappings. A technique for identity resolution and pay-as-you-go data integration is explained. In order to provide a strong degree of assurance, the authors subject the model to certain real world entities that might form part of a global dataspace.

It is discussed that dataspace entities in principal can be modelled to take a hybrid relationship which combines both hierarchical and network model in order to be sufficient. An example is that a page belongs to a document but it is also true to state that a page belongs to a website. Moreover, a website may also consist of documents [24]. Fig. 5. illustrates the design of the Hybrid model.

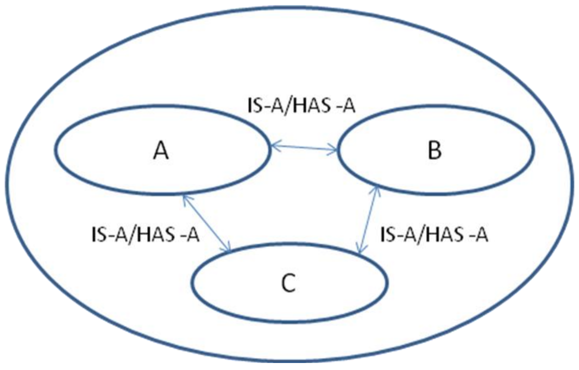

Figure 5. The Hybrid Dataspace Model (Adopted from Shibwabo, Wanyembi and Ateya [24]).

A critical element to understand the hybrid model is the domain. A dataspace will typically represent a specific domain thereby providing a higher degree of sense in what is being represented. A dataspace can be represented as a Set and the participants in a dataspace as the Set Elements. This is computationally feasible due to the fact that there exists a practical programming implementation defined as set<Key, Compare, Alloc>. Where:

- Key: The set’s key type and value type. This is also defined as set::key_type and set::value_type;
- Compare: The key comparison function, a Strict Weak Ordering whose argument type is key_type; it returns true if its first argument is less than its second argument, and false otherwise. This is also defined as set::key_compare and set::value_compare;
- Alloc: The set’s allocator, used for all internal memory management.

This paper conducts an extensive analysis, design and further implementation of a dataspace support system. A key observation/ issue in the proposed approach is the lack of guarantee for scalability. There is also need to develop an intuitive interface for users to interact with the proposed platform.

IV. SUMMARY AND CONCLUSIONS

In summary, the problem of data integration has been investigated for years with the aim of providing end users with integrated access to data sets that reside in multiple sources and are stored using heterogeneous representations. In order to solve this problem, dataspaces have been proposed to provide a cheaper and more flexible way of managing the data integration problem. With dataspaces, not only is integration provided incrementally but the aim to provide for data coexistence is more simplistic. Dataspaces target to model, query, and manage relationships among disparate data sources. They further eliminate the need for up front semantic integration which is common in current integration approaches that have generally proven to be complicated and expensive. The data integration problem is often described as the need to manage relationships among disparate data sources.

The development of dataspace support platforms (DSSPs) is an interesting research area that is still at its infancy. There are various challenges in the development of DSSPs including the identification of participants and their relationships, learning and discovery, query modeling, reusing human attention among other challenges. Additional concerns that are profound regarding dataspace implementation include scalability concerns due to the growth of data and the development of security model to deal with the intersection and access of dataspaces. Additional work need to be done on the development of an intuitive interface for users interacting with a dataspace support system. More theoretical work has been presented in literature regarding the requirements and design of dataspaces. Very little literature exists on a working implementation as per the initial dataspaces vision and later principles of a dataspace described in literature. There is need for research on the data structures and Algorithms that support dataspaces. This way, the dataspace vision can become a reality.

Although notable theoretical work has been done towards the requirements and design of a dataspace, the development of these concepts in a real implementation that addresses all the indicated challenges is not complete. The successful development and implementation of dataspaces is expected to solve to a great extent the data integration problem which is prevalent in organizations and individual desktops. This way, more time will be used to perform value adding tasks as opposed to using the time to solve current data integration issues.

ACKNOWLEDGEMENT

The authors like to thank Strathmore University for the PhD Scholarship. This work was supported in part by a grant from The National Council for Science and Technology (NCST) in the Republic of Kenya grant reference number NCST/ST&I/RCD/4th Call PhD/213 under the Science, Technology and Innovations Grant fund (ST&I). The authors would like to thank NCST.

REFERENCES

[1] K. Belhajjame,N. Paton,S. Embury, Feedback-Based Annotation, Selection and Refinement of Schema Mappings for Dataspaces, ACM EDBT 2010, March 22–26, 2010, Lausanne, Switzerland, 2010.
[2] M. Franklin, A. Halevy, and D. Maier, “Principles of dataspace systems”, Proc. of Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2006), ACM Press, p. 1-9., ISBN 1-59593-318-2.
[3] M. Franklin, A. Halevy, D. Maier, From databases to dataspaces: A new abstraction for information management, ACM SIGMOD Record 34 (4) , 2005, 27-33.
[4] P. Ziegler, K. Dittrich, Data Integration — Problems, Approaches, and Perspectives, Springer, Berlin Heidelberg, 2007.
[5] A. Marcos, S. Vaz, D. Jens, B. Lukas, “Intensional associations in dataspaces,” In: ICDE 2010, (2010)
[6] S. Song, L.Chen, M. Yuan, "Materialization and Decomposition of Dataspaces for Efficient Search," Knowledge and Data Engineering, IEEE Transactions on , vol.23, no.12, pp.1872,1887, Dec. 2011
[7] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. Theor. Comput. Sci., 336(1):89–124, 2005.
[8] M. Lenzerini. Data integration: A theoretical perspective. In L. Popa, editor, PODS, pages 233–246. ACM, 2002.
[9] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. VLDB J., 10(4):334–350, 2001.
[10] X. L. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. VLDB J., 18(2):469–500, 2009.
[11] A. Bonifati, G. Mecca, A. Pappalardo, S. Raunich, and G. Summa. Schema mapping verification: the spicy way. In EDBT, pages 85–96. ACM, 2008.
[12] A. Gal. Why is schema matching tough and what can we do about it? SIGMOD Record, 35(4):2–5, 2006.
[13] R. J. Miller, L. M. Haas, and M. A. Hernández. Schema mapping as query discovery. In VLDB, pages 77–88, 2000.
[14] L. Xu and D. W. Embley. A composite approach to automating direct and indirect schema mappings. Inf. Syst., 31(8):697–732, 2006.
[15] L. Chiticariu and W. C. Tan. Debugging schema mappings with routes. In VLDB, pages 79–90. ACM, 2006.
[16] C. J. van Rijsbergen. Information Retrieval. Butterworth, 1979.
[17] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. Knowl. Eng. Rev., 18(2):95–145, 2003.
[18] J. Dittrich, M. Salles, iDM: A Unified and Versatile Data Model for Personal Dataspace Management, VLDB ‘06, September 1215, 2006, Seoul, Korea.

[19] H. Ning, T. Wang. Constructing a Dataspace Based on Metadata and Ontology for Complicated Scientific Data Management, IEEE, 2007.

[20] Gruber TR., "A translation approach to portable ontology specifications", Technical Report, KSL 92-71, Knowledge System Laboratory, 1993

[21] I. Elsayed and P. Brezany. Towards Realization of Dataspaces, Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA'06), IEEE, 2006

[22] F. Berman, A. J. G. Hey, and G. Fox. Grid computing: Making the global infrastructure a reality. John Wiley & Sons, 2003.

[23] I. Foster. Globus toolkit version 4: Software for service-oriented systems. IFIP International Conference on Network and Parallel Computing, Springer- Verlag LNCS 3779, pp 2-13, 2005.

[24] B. K. Shibwabo, G. N. Wanyembi and I. L. Ateya (2012). Modelling Data space Entity Association using Set Theorem. Computer Technology and Applications, Vol. 3, No. 6, 2012.