View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Archivio istituzionale della ricerca - Università di Modena e Reggio Emilia

No users no dataspaces! Query-driven dataspace orchestration*
(position paper)

George H. L. Fletcher[1] and Federica Mandreoli[2]

[1] Eindhoven University of Technology
The Netherlands
g.h.l.fletcher@tue.nl
[2] University of Modena and Reggio Emilia
Italy
federica.mandreoli@unimore.it

Abstract. Data analysis in rich spaces of heterogeneous data sources is an increasingly common activity. Examples include querying the web of linked data and personal information management. Such analytics on dataspaces is often iterative and dynamic, in an open-ended interaction between discovery and data orchestration. The current state of the art in integration and orchestration in dataspaces is primarily geared towards close-ended analysis, targeting the discovery of stable data mappings or one-time, pay-as-you-go ad hoc data mappings. The perspective here is dataspace-centric.
In this paper, we propose a shift to a user-centric perspective on dataspace orchestration. We outline basic conceptual and technical challenges in supporting data analytics which is open-ended and always evolving, as users respond to new discoveries and connections.

1 The vision

In many contemporary data management scenarios, users are faced with large collections of independent heterogeneous data sources with which they initially have limited understanding of the structure and semantics. The technical solutions for data sharing are by now quite mature: RESTful API's, linked data standards, and so forth. What remains fundamentally in querying over such spaces of data sources are issues of orchestrating the sources for querying, that is, aligning and exchanging data between sources, towards resolving an information seeking task.

Some example use cases of this scenario are: querying the web of linked data [10], Personal Information Management (PIM) systems [1], and exploratory data analysis [5]. Much of the basic ingredients towards supporting users in these scenarios is by now well understood, coming from the rich literature on data

* This article is based upon work from COST Action KEYSTONE IC1302, supported by COST (European Cooperation in Science and Technology)

integration, (p2p) dataspaces, and data exchange [7, 14, 8, 9]. Of these solutions emerging in the last years, all are aimed at a common objective that is answering to the need of a unified search over the full spectrum of relevant knowledge.

The management of mappings for orchestration of data sources in this context has been almost exclusively from a data-centric perspective. Indeed, in the current state of the art, information and orchestration of dataspaces is primarily geared towards close-ended analysis, targeting the discovery of stable data mappings or one-time, pay-as-you-go data mappings.

A shift in perspective In contrast to this, analytics in the dataspace scenarios discussed above is often iterative and dynamic, where new query results inform and guide further analysis, in an open-ended interaction between discovery and data orchestration. Indeed, the world is always changing, people are all different, people themselves are always changing. Data semantics is in the eye of the consumer. Hence, schemas/ontologies are always idiosyncratic. Furthermore, idiosyncrasies are always evolving. The data engineering community (and, more generally, the computer science research community) has been grappling with this dynamism, this flux in user-driven data usage, since the founding of the field [11].

In this light, close-ended analysis is not the norm, but rather an exceptional activity. In fact, in the absence of users and their information seeking activities, there is no need for orchestrating data sources, and hence no dataspaces. We capture these observations in the slogan *No users no dataspaces!* We are motivated by this to propose a shift of perspective, from data-centric orchestration to user-centric orchestration. This shift of perspective essentially changes the role of mappings that are no longer aimed to support data exchange or data integration application scenarios but rather to contribute to users satisfaction in their information seeking activities. From a data management point of view, the primary visible expression of users are their queries (i.e., the embodiment of a fleeting view and information need in the world). Hence, we start from the perspective of the adhoc query, which we then try to satisfy in the dataspace. In this process of satisfaction, we often need to define and refine *mappings* between data sources and expose the user to the data thereby discovered for their feedback and possibly continued reformulation of their queries. The main novelty thus lies in a different approach to the issue of mapping management that is triggered by the specific user and his/her query and targets to his/her satisfaction. In this way, we perform an important step towards realizing the vision of a fully fledged pay-as-you-go information integration approach for analytics on dataspaces.

Towards realizing this shift And of course, users do not live in isolation, but rather in a diversity of rich overlapping communities. Therefore, all of the effort involved in defining and refining mappings is shared by individuals, within and across communities, across time granularities. Pareto's principle[3] tells us that (1) there are many widely shared commonalities in information needs, and (2) information needs are almost always highly focused and topical, not requiring orchestration at the level of traditional data integration systems. This gives us

[3] `https://en.wikipedia.org/wiki/Pareto_principle`

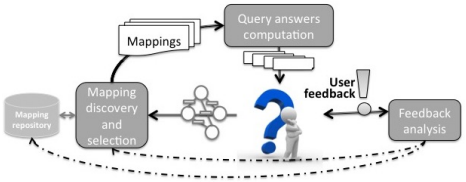

Fig. 1. The QyX query-driven dataspace orchestration process. Solid lines are used to depict the process while dashed lines show the feedback analysis impact.

hope as engineers, towards the design and implementation of effective and efficient user-centric dataspace solutions. Our goal in this paper is to sketch the first steps in this direction by proposing a query-driven dataspace orchestration framework and by discussing the main challenging issues that should be addressed in this context.

2 Query-driven dataspace orchestration

By query-driven dataspace orchestration we mean open-ended dynamic dataspace orchestration, interactively driven by ad-hoc user requests and their interaction with the views induced by past queries and query-refinement and reformulation.

The data modeling abstraction we adopt to represent a dataspace is as lightweight as possible, for the goals of our study. To this end, it is (a) *fully decentralized*, thereby bridging, on the one hand, existing dataspace models that usually rely on a single mediated view [7] and, on the other hand, P2P approaches for data sharing [9]. A dataspace is therefore a collection of autonomous data sources locally connected through an interlinked collection of semantic mappings used to answer past user requests; (b) *schema-flexible*, in the sense that data sources can be schemaless and users do not need to know the complete and exact structure of the data to query it; (c) *based on data exchange*, in that when answering user queries the data source includes data objects that reflects a given set of mappings between the data source and the dataspace [8].

In this context, for the sake of simplicity we assume that an ad-hoc user request is expressed over one data source. The goal of answering the request satisfactorily induces a dataspace orchestration that aims at aligning the dataspace to the specific request through the interactive execution of three steps: mapping discovery and selection, query answer computation, feedback analysis. Figure 1 depicts the process that can stop when the user is satisfied with the delivered answers or when all the alternative answer sets have been already shown to the user. It relies on a mapping repository that stores the mappings used to answer past requests. The update of the repository is part of the process.

In the following, we discuss the key ideas of the process using a running example in a PIM scenario focused on music and composers. We assume that

the dataspace is made up of three data sources: a small portion of a social graph containing information about what user's friends like and two knowledge graphs, one focused on composer biographies and the other one on schools of music and art. Data are represented as triples according to the flexible entity-based modeling abstraction typical of dataspaces [7]. In this simplified example we assume that the data sources are aligned in their objects. For instance $(john, likes, mozart)$ and $(likes, type, vote)$ are two triples of the first graph data source G_1 while $(mozart, genre, classical)$ and $(classical, period, 17thCentury)$ belong to the second and third data source, G_2 and G_3, respectively. Let us assume that the first query issued by the user on G_1 asks for the period user's friends are fan of. This query can be expressed in a conjunctive pattern matching form as follows, where $?x$,$?y$, and $?z$ are variables:

$$Q_1 : (?x, periodFanOf, ?z) \leftarrow (?x, likes, ?y), (?y, period, ?z)$$

2.1 Query-driven mapping discovery and selection

Mapping discovery is the core step of the framework and arises many interesting research issues that are discussed in Sec 4. It is query-driven in that it is triggered by the query and aims at discovering mappings in the dataspace and refining existing mappings that are necessary to solve the query itself. For instance, G_1 has not enough information to solve Q_1 or, equivalently, Q_1 can be solved only partially on G_1. Then, the system implements an algorithm that delivers at each step a set of candidate mappings that can be used to totally solve the query.

The algorithm we envision is based on the principle that the match is "partial" by virtue of missing data. For instance for Q_1, $(john, likes, mozart)$ matches $(?x, likes, ?y)$ whereas we assume that there is no triple matching the pattern $(?, period, ?z)$. Therefore, the answer set is not empty if the algorithm finds at least one mapping that generates the missing triple $(mozart, period, w)$ where w is any value. To this end, the algorithm relies on two elements that can be derived from each partial match: the values associated with the bounded query variables, to extract examples from the dataspace under structural indistinguishability [15], and the unmatched query statement, to generate the target of the missing mappings. With reference to the example above, the algorithm extracts the triple $(mozart, genre, classical)$ from G_2 and uses the pattern $(?y, period, ?z)$ to deliver the mapping that define the pattern $(?y, period, ?z)$ on G_1 as the joining of the pattern $(?x, genre, ?y)$ from G_2 with $(?y, period, ?z)$ from G_3:

$$m_a = G_1 : (?x, period, ?z) \leftarrow G_2 : (?x, genre, ?y), G_3 : (?y, period, ?z)$$

This mapping extends G_1 with triples of the form $(w_1, period, w_2)$, where w_1 and w_2 are values, also including $(mozart, period, 17thCentury)$ that is used to compute the answer $(john, periodFanOf, 17thCentury)$ to query Q_1.

During this step, the system capitalizes on past orchestration work by mixing the actual discovery task with a focused selection of mappings from the mapping repository. For instance if m_a is available in the repository and G_1 receive a query

asking for the century the user's friends are interested in:

$$Q_2 = (?x, CentOI, ?z) \leftarrow (?x, likes, ?y), (?y, century, ?z)$$

then m_a can be used to build mapping m_b that renames *period* into *century*

$$m_b = G_1(?x, century, ?y) \leftarrow G_1 : (?x, period, ?y)$$

and that would bring the answer $(john, centOI, 17thCentury)$.

2.2 User-oriented strategies for mapping management

The main observation that motivates our proposal is that data semantics is in the eye of the consumer. In our vision different users seeking the same information or the same user seeking the same information at different times are exposed to different satisfaction processes and thus dataspace orchestrations, depending on their backgrounds, interests and current situations.

To this end, the user's profile and situational context are represented as query metadata and exploited for tailored data analysis [12]. At the same time, new query results can inform and guide users for further analysis. We envisage a fruitful interaction with the user that can be involved in query reformulation/refinement tasks and the explorations of intermediate as well as complete query answers. The system then exploits the user behaviour and feedback to annotate mappings with metrics measuring their fitness to user expectations [4].

User-oriented mapping management strategies play a fundamental role in dataspace orchestration for effectiveness and efficiency reasons. As to mapping discovery, it is worth noting that the space of the candidate mappings that can be derived from an issued query is usually very large as it relies on the whole dataspace. User-oriented strategies can guide the scanning of such a wide space toward the mappings that best fit user expectations according to the knowledge the system gained from query metadata and past iterations. As to mapping selection, the mapping collection is indexed over the different user-oriented features of interest such as the queries for which mappings were used and the related fitness metrics, the profiles of the users who submitted such queries and the related contexts. Then, different mapping selection strategies can be adopted.

For instance, if the system follows a strategy that aims to limit mapping discovery efforts, Q_2 can be first evaluated by defining mapping m_b through m_a. Then, if the user is not satisfied with the delivered answers, the system can decide to drive the discovery toward alternative $(?y, century, ?z)$ definitions, e.g. that relate composers with the 20th-century music because, thanks to the interaction with the user, it has understood this is the *century* interpretation user has in mind.

3 Two case studies

In this section we discuss two case studies of data analysis that show different user-system interaction models and give an hint of how the query-driven dataspace orchestration framework we propose can be of support to the user task.

3.1 Case study 1: Exploratory data analysis

Exploratory data analysis helps users to make sense of very big datasets. In [5] it is described as the step-by-step "conversation" of a user and a system that "help each other" to refine the data exploration process, ultimately gathering new knowledge that concretely fulfills the user information needs. Exploratory analysis tools address both data scientists, such as biologists, that have a deep knowledge of the domain of interest and clear and peculiar information needs and data enthusiasts, such as journalists, that want to analyze the data to achieve new and essential knowledge of the domain of interest.

The proposed query-driven dataspace orchestration paradigm can be straightforwardly integrated in the data exploration loop where it allow the system to overcome the logic of one-size-fits all. In this context, the thorough conversation between the user and the system represents the main source for dataspace orchestration to understand the user information needs and purposes. We envision a dataspace orchestration process that, when the conversation starts, leverages the initial knowledge of the user and the user profile for a focused mapping discovery and selection. Then, it will gradually adapt to the new knowledge the user will gain while using the system as well as his/her changes of perspective and interest. For instance, in case of a data enthusiast user, mapping discovery and selection could be initially driven toward jargon-free data objects that would bring easily understandable answers. Then the process will move toward more specialized data sources that could help the user to gain a deeper knowledge of the domain of interest. In this way, each user will have his/her own view of the dataspace that will evolve over time according to his/her level of knowledge and topics of interest.

3.2 Case study 2: Data analysis in dynamic contexts

A significant involvement of the user during the interpretation process is not adequate in dynamic contexts, in which the interaction is often hampered by the communication means and by the need to quickly obtain answers [6]. This is the case, for instance, of a mobile user that would like to find answers to complex requests involving his/her current position.

This kind of requests often concerns geo-referenced and time-variant information as well as crowdsourced data that need to be integrated with strongly correlated and semantically complex data (e.g., Linked Open Data) and unstructured data, or data with a simple and defined structure. The dataspace of interest is therefore made up of highly heterogeneous and dynamic data sources.

In such a dynamic context, standard mapping management approaches could bring to costly and useless integration results because of the rapid changes that characterize the considered scenario.

Catania et al. [6] identify three coordinates relevant in this context: (a) user profile and request context; (b) data and processing quality; (c) similar requests repeated over time. For example, thanks to (a) and (b), it is possible to prefer synthetic and timely answers sacrificing the quality of result in the case of a

user on the move or in an emergency situation. The last coordinate, (c), is very common in dynamic contexts and can be used to limit response times and interpretation errors. It occurs, for example, during or after an exceptional event (environmental emergencies or flash mobbing initiatives), in the context of users belonging to the same community or that are in the same place, possibly at different times. The information needs are widespread among different users, because induced by the event, the interests of the community, and the place, respectively.

Our framework for query-driven dataspace orchestration can benefit from these three coordinates to focus mapping management efforts over such information that can contribute to produce satisfactory answers. In particular, profiled query patterns, that is synthetic representations of past requests associated with the corresponding user profiles and contexts, can be used to index the mapping repository for effective mapping searches. Moreover, the dynamism requirements of the submitted request can be exploited both for mapping selection and to drive mapping discovery toward the candidate mappings that best meet the request time.

4 Open challenges and future research directions

Our proposal addresses an old and pressing problem from a new perspective. Although, we can borrow some solutions from past research works there are still several substantial research issues which need to be addressed to complete the picture. We list some of the main issues below.

Although some mapping discovery approaches via data examples have been proposed [2], the non-trivial challenges highlighted by a query-driven perspective on mapping discovery indicates fresh perspectives on research. Indeed, in this case, the discovery is driven by queries, instead of data, and the aim is to find sets of candidate mappings that extend partial matches to total ones. This novel mapping discovery paradigm requires a rigorous in-depth theoretical study: the introduction of a formal notion of candidate mapping followed by a query-driven mapping-discovery algorithm that is sound and complete, i.e. it must deliver all and only the candidate mapping sets for a given request. In this context, it is also interesting to study the impact of different graph query languages on the theoretical framework. To this end, an interesting paper is [3] that studies problems of materializing solutions and query answering for these languages.

The problem of checking the existence of total or partial matches as well as their computation arose in various steps of the process. Although some papers already provide interesting solutions [2], they cannot be adopted as they are because of the relationship between the query to be matched and the network of mappings connecting data sources that has never been addressed before.

Users are not willing to wait query answers for a long time and, then, a system built on the proposed paradigm has strict time constraints that are unknown to standard mapping management systems. As a consequence, both the above research problems need efficient and scalable solutions. From an engineering

point of view, the above algorithms can largely benefit from structural indexes and histograms similar to the ones proposed in [16, 15] as well as an effective dataspace representation as an easy-to-navigate graph. Then, specific tasks, such as object alignment, can be faced through on-the-fly techniques [13].

Finally, as to user involvement, we see two core issues that require innovative solutions in the engineering field. Indexing and selection techniques are necessary for the mapping repository. Here, the main questions to be addressed are how to reuse previous mapping collections and to compare/score alternative mappings. As shown in Sec. 3, these issues require solutions specifically tailored to the application scenario. As far as the user feedback is involved, instead, we first need to introduce effective feedback elicitation approaches, similarly to the ones proposed in [4]. Then, we should propose approaches that translate the feedback and behaviour of the user on query results into some kind of mapping annotation useful for mapping selection and discovery.

References

1. Abiteboul, S., Marian, A.: Personal information management systems. In: EDBT Tutorial. Brussels (2015)
2. Alexe, B., ten Cate, B., Kolaitis, P.G., Tan, W.C.: Designing and refining schema mappings via data examples. In: Proc. of SIGMOD. pp. 133–144 (2011)
3. Barceló, P., Pérez, J., Reutter, J.L.: Schema mappings and data exchange for graph databases. In: Proc. of ICDT. pp. 189–200 (2013)
4. Belhajjame, K., et al.: Feedback-based annotation, selection and refinement of schema mappings for dataspaces. In: Proc. of EDBT. pp. 573–584 (2010)
5. Buoncristiano, M., et al.: Database challenges for exploratory computing. SIGMOD Record 44(2), 17–22 (2015)
6. Catania, B., Guerrini, G., Belussi, A., Mandreoli, F., Martoglia, R., Penzo, W.: Wearable queries: adapting common retrieval needs to data and users. In: Proc. of DBRank (co-located with VLDB). pp. 7:1–7:3 (2013)
7. Dong, X., Halevy, A.Y.: Indexing dataspaces. In: Proc. of SIGMOD. pp. 43–54 (2007)
8. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theoretical Computer Science 336(1), 89–124 (2005)
9. Halevy, A.Y., Ives, Z.G., Suciu, D., Tatarinov, I.: Schema mediation for large-scale semantic data sharing. VLDB J. 14(1), 68–83 (2005)
10. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web, Morgan & Claypool Publishers (2011)
11. Kent, W.: Data and reality. North Holland, Amsterdam (1978)
12. Koutrika, G., Ioannidis, Y.E.: Personalizing queries based on networks of composite preferences. ACM TODS 35(2) (2010)
13. Papadakis, G., Koutrika, G., Palpanas, T., Nejdl, W.: Meta-blocking: Taking entity resolution to the next level. IEEE TKDE 26(8), 1946–1960 (2014)
14. Penzo, W., Lodi, S., Mandreoli, F., Martoglia, R., Sassatelli, S.: Semantic peer, here are the neighbors you want! In: Proc. of EDBT. pp. 26–37 (2008)
15. Picalausa, F., Fletcher, G.H.L., Hidders, J., Vansummeren, S.: Principles of guarded structural indexing. In: Proc. of ICDT. pp. 245–256 (2014)
16. Scholl, T., et al.: Hisbase: Histogram-based P2P main memory data management. In: Proc. of VLDB. pp. 1394–1397 (2007)