Data Exchange

Evgeny Pozdeev, Fernando Stefanini
Big Data Management and Analytics

Introduction

Data Exchange characterizes the transfer of data between two different schemas, source and target, usually trying to ensure that same information found in the source schema is represented in the target schema as trustworthy as possible to the original. In order to do so, usually, the process can be split into two conceptual steps: creating the mapping between the schemas and doing the transformations necessary to convert the data from the source to the schema of the target [1].

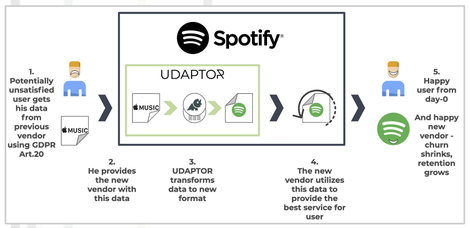
It's intuitive to compare it to the creation of ETL's (extract-load-transform) processes (generally associated with data warehousing). In these scenarios, the usual context is given by trying to put together data from multiple data sources into a big monolithic schema that allows OLAP (Online analytical processing) queries in order to derive aggregate results that could lead to an insightful understanding of the information. However, we also would like to point that more up to date contexts can also be related to it.

Use Cases

Article 20 of the General Data Protection Regulation, for instance, ensures that any individual residing in the European Union has the right of data portability. In short, it states that a person shall be able to request his/her personal data from a controller authorized to collect it and use it to port to another controller [2]. Here, although the regulation doesn't specify that means by which the transfer must be done, clearly we apply the concept of data exchange between the two providers aiming to make possible for people to exercise their data portability right.

Picture 1 – General Data Protection Regulations.

Although intuitive, these examples are simplistic and enough to raise questions as if data exchange is the only mechanism to cope with these scenarios, as well as being the most suited. Given that, we propose a discussion on why it's fitted using Business Intelligence needs as an example.



Picture 2 – Example of a person exercising the Data Portability Right given by GDPR with the help of UDAPTOR.

Moreover, another discussion associated with data exchange in the current status quo is the ability to automate it in order to use it in a scalable fashion [3]. So the sections after that will address a discussion over two different implementations on data exchange, one traditional and one probabilistic approach, in order to present the current feasibilities of applying automated data exchange in a real-life scenario.

Data Exchange in Business Intelligence 2.0

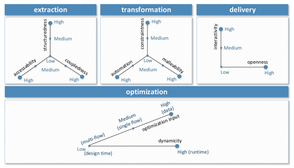
In the reviewed paper [3] authors conduct a survey on technologies focused on Data Intensive flows, where the last one is defined as a critical process in today's Business Intelligence systems, especially focusing on the next generation BI. The wide concept of Data Intensive flows was analyzed from the perspective of 2 scenarios: ETL and ETO. Since both components conceptually represent the same thing, the authors propose to conduct a comparative analysis using a unique set of metrics per each stage of those concepts (Table 1).

	Extraction	**Transformation**	**Delivery**
ETL	*Accessibility, structuredness, coupledness*	*Automation, constraintness, malleability*	*Interactivity, openness*
ETO			
	Optimization input, dynamicity		
	Optimization		

Table 1 - Stages of ETL and ETO (in bold) and their characteristics (in italics) under which the comparative analysis was conducted[3].

Despite the fact that ETL and ETO were only 2 scenarios under analysis, authors often relate to Data Exchange while elaborating on certain stages, especially the Transformation and Delivery stage. The definition of Data Exchange that was given in the Introduction fits into "handling heterogeneity of the data" part of Intensive Data flows, and in this field, Data Exchange was studied to give an opportunity to a user to query data over Target schema without referring to Source schema[4]. In addition to that, often relations to Data Exchange can be justified with the fact that research[5] shows that the problem of Data Exchange is conceptually close to the traditional ETL problem.

For one of the Transformation characteristics, constrainedness authors gave 'High' mark. It is justified by the fact that together with schema mappings constraints target schema also put additional constraints on the transformation process: the derived instance has to satisfy restrictions defined by the target schema. For other Transformation characteristics - Automation - Data Exchange was also given 'High' mark due to the existence of an automation tool. The tool is called Clio and it will be analyzed more deep in the next section, but the main reason why it allows to put 'High' Automation mark is the ability to generate schema mappings without making any initial assumptions about relations between Source and Target schemas.

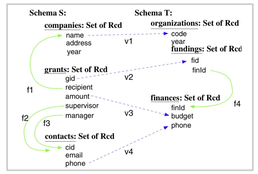


Picture 3 – Dimensions for characteristics used in comparative analysis. Taken from [3].

Talking about the next stage - Data Delivery - authors mentioned Data Exchange in Interactivity and Openness. For Interactivity Data Exchange was introduced with one fundamental problem: incompleteness of data and/or schema, which leads to a theoretically infinite number of valid solutions. In order to overcome the problem of intractability, the concept of Universal solution was proposed[6], it uses homomorphism to any other possible solution and helps to get rid of that infinite number of possible solutions. For the second characteristics of Data Delivery - Openness - the mark 'Low' was given due to the fact that for the theoretical problem of Data Exchange Close World Assumption is more suitable: for any query, the expected answer has to strictly rely on data that was transferred from Source to Target schema.

Actual implementation: Clio

The article under review named "Clio: Schema Mapping Creation and Data Exchange"[8] and represents the description of one of the key contributions into joint Clio project between IBM Almaden Research Center and the University of Toronto. The main goal of the project is a radical simplification of information integration by means of automating and managing the conversion of data between representations (schemas and/or data formats). The paper under review focuses mostly on 2 parts of that problem: mapping 2 heterogeneous schemas and the usage of that mapping for data exchange.



Picture 4 – Example of Source and Target schemas with mapping. Taken from [3]

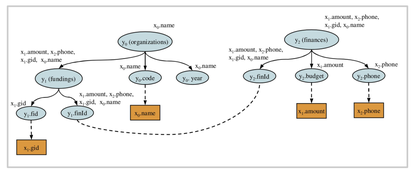
Requirements for Clio development include but are not limited to:

- There no assumption made on the relationship between schemas or on the schema itself;
- Creation and usage of mappings have to happen on a different level of granularity;
- Mappings creation algorithm has to be incremental.

However, there are several restrictions/assumptions under which Clio can be used:

- Mapping creation between any 2 given schemas is not commutative;
- (In the paper, it is stated) the problem of creating correspondences is out of scope.

The proposed algorithm uses 3 levels of data representation which corresponds to 3 logical steps: deriving association, creating mappings and building query to perform Data Exchange. Besides that, within each step, there are multiple sub-levels (e.g. Partitioned Normal Form, query graph, etc.).



Picture 5 – Derived query graph. Taken from [3].

The process that affects most on resources (time, computational) is the chase phase. In general, it is known that the chase may not terminate, so Clio uses a special case of the chase with additional constraints (schemas represent in the weakly-acyclic set form). In this case, the number of chase steps is polynomial, but each of the chase steps can be exponential at worst which almost never happens.

The authors of the article state that there are 2 main innovations that their algorithm brought to Data Exchange research:
Usage of Skolem technique allows performing grouping in the target schema
Ability to identify and merge data that is equivalent, but redundant (was generated by different mappings)

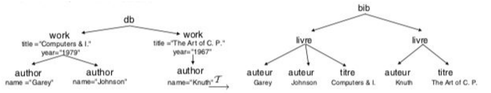
In addition to that, Clio is the first tool that used schema mapping queries to represent a relationship between heterogeneous data sources, which lead to a new paradigm in Data Exchange.

Theoretical approach: Probabilistic Data Exchange

In "Approximate Data Exchange"[11] the authors propose a more theoretical approach to the problem, making use of probabilistic methods in order to be able to relax the original data exchange problem. The steps involved are:

- Defining a transducer that will try to convert an instance of the source schema to an instance of the target schema and running it over multiple instances of the source schema;
- Using known techniques to determine how far is the transformation from the target schema;
- Using Property Testing to check whether there is a transformation that has a distance less than ε to target schema, which will represent the solution to the mapping problem that can now be applied to all other instances.

The approach used for transducer is a Top-down Tree Transducer[12], which leverages well on the use of Regular Expression to define the schema language for many cases but can have its run time drastically affected if a very expressive regular expression is needed to describe the target.



Picture 6 - Example of the transformations made by a transducer. Taken from [11]

In the paper, it is proposed that an "Edit Distance with Moves" [13] operation can be applicable to calculate the distances. While that is true, in practice and differently for the classical edit distance problem that can be solved with dynamic programming, this version has an NP-Complete nature on the size of the strings or requires the usage of a greedy algorithm to get to a fair approximation of the actual result.

The last step is to assess whether a valid mapping to be applied to the other instances was created by means of using Property Testing. Here we highlight that although Property Testing has the nice feature of its complexity being able to depend solemnly on ε, as any other probabilistic approach it requires finding an optimal value for it.

If considered that in the real world the cases where the limitations for each one of the approaches could yield a massive drawback in the final performance of the methodology might not happen, the suggested solution would present a performance that is polynomial in the number of entities that needs to be mapped from one schema to another. However, to have a clear understanding if that would still account for a runtime that is practical in a real-world application, either further research on each one of the steps needs to be done, or benchmarks need to be performed on top of actual Data Exchange Scenarios.

One important thing to keep in mind is that the most innovative aspect of the solution is abstracting the Data Exchange problem into smaller problems that can be reduced to known and studied challenges that we already face nowadays. One of the ways that that favors the solution, other than the already done research on these topics, is the fact that it could still continue on benefiting from further discoveries in each one of them. Moreover, each one of the steps is performed in an independent way, such that replacing one of them for a more robust or fitted technique could probably be done in the future.

Conclusion

In this article we aimed to discuss the relevance of Data Exchange, how it's inserted in the current context of massive data flows and the feasibility of its implementation in an automated way. We started by introducing the topic and its applications and went on showing other researches that believe it is fit for Business Intelligence 2.0.

Moreover, we showed two different approaches to implementing a real-life data exchange system. While the first one, which we called a traditional algorithmic approach, has already been implemented, it requires a few manual steps that could make it possibly hard to scale when dealing with a rapidly growing / changing scenario. On the other hand, it can automatically transform data even considering if it's composed of many fields in the source schema.

Meanwhile, the probabilistic approach, although not yet implemented, opens up the possibility of automating the whole pipeline of data exchanging. The main caveats here are the uncertainty that the assumptions and limitations bring, considering that most of the techniques employed are shown to have linear or polynomial running complexity in most cases, but there is no guarantee that this will still account for a feasible real life execution time. Also, it's important to remember that in specific cases the probabilistic approach can't guarantee linear or polynomial running times as well.

Mainly, we would like to highlight how our analysis showed Data Exchange as still relatively undeveloped field both in terms of implementation and theoretical approaches. It's important to acknowledge that advances could probably come by having more theoretical work being applied in scenarios that are consistent with those we face in the current state of big data management. Not only we would have a better understanding of its feasibility, but we would most likely have better insights on how to improve the solutions that are already implemented.

Bibliography

[1] Ronald Fagin, Ronald Fagin, and Ronald Fagin, Phokion G. Kolaitis, Lucian Popa. 2005. Data exchange: getting to the core. ACM Trans. Database Syst. 30, 1 (March 2005), 174-210. DOI: https://doi.org/10.1145/1061318.1061323
[2] General Data Protection Regulation. Chapter III. Article 20. Right to data portability. Available at: https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:32016R0679
[3] Jovanovic, Petar & Romero, Oscar & Abelló, Alberto. (2016). A Unified View of Data-Intensive Flows in Business Intelligence Systems: A Survey. 10.1007/978-3-662-54037-4_3.
[4] Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
[5] Vassiliadis, P.: A survey of extract-transform-load technology. IJDWM 5(3), 1–27 (2009)
[6] Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data Exchange: Semantics and Query Answering. Theoretical Comput. Sci. 336(1), 89–124 (2005)
[7] Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data Exchange: Semantics and Query Answering. In: ICDT. pp. 207–224. Springer (2003)
[8] Fagin R., Haas L.M., Hernández M., Miller R.J., Popa L., Velegrakis Y. (2009) Clio: Schema Mapping Creation and Data Exchange. In: Borgida A.T., Chaudhri V.K., Giorgini P., Yu E.S. (eds) Conceptual Modeling: Foundations and Applications. Lecture Notes in Computer Science, vol 5600. Springer, Berlin, Heidelberg
[9] Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. J. ACM 31(4), 718–741 (1984)
[10] Abiteboul, S., Bidoit, N.: Non-first Normal Form Relations: An Algebra Allowing Data Restructuring. J. Comput. Syst. Sci. 33, 361–393 (1986)
[11] de Rougemont, Michel & Vieilleribière, Adrien. (2007). Approximate Data Exchange. 4353. 44-58. 10.1007/11965893_4.
[12] W. Martens and F. Neven. Frontiers of tractability for typechecking simple XML transformations. In ACM Principles on Databases Systems, pages 23–34, 2004.
[13] Shapira D., Storer J.A. (2002) Edit Distance with Move Operations. In: Apostolico A., Takeda M. (eds) Combinatorial Pattern Matching. CPM 2002. Lecture Notes in Computer Science, vol 2373. Springer, Berlin, Heidelberg