Built-up integration: A new terminology and taxonomy for managing information on-the-fly

Maria Franciscatto, Luis Carlos Erpen de Bona, Celio Trois, Marcos Didonet del Fabro

Journal of Information and Data Management, 2023, 13:1, doi: 10.5753/jidm.2023.XXXXXX

Built-up Integration: A New Terminology and Taxonomy for Managing Information On-the-fly

Maria Helena Franciscatto [**Federal University of Paraná** | ***mhfranciscatto@inf.ufpr.br***]
Luis Carlos Erpen de Bona [**Federal University of Paraná** | ***bona@inf.ufpr.br***]
Celio Trois [**Federal University of Santa Maria** | ***trois@inf.ufsm.br***]
Marcos Didonet Del Fabro [**Universitè Paris-Saclay, CEA, List** | ***marcos.didonetdelfabro@cea.fr***]

Received: DD Month YYYY • **Accepted:** DD Month YYYY • **Published:** DD Month YYYY

Abstract Obtaining useful data to meet specific query requirements usually demands to integrate data sources at query time, which is known as on-the-fly integration. Currently, many studies address this concept by discovering useful data sources in an ad-hoc manner, and merging them for providing actionable information to the end user. This set of steps, however, lack a standardization in their identification, since they are described in the literature under many different names. Hence, without an unified nomenclature and knowledge organization, the development in the area may be considerably impaired. This paper proposes a novel term called Built-up Integration aiming at knowledge regulation, and a taxonomy for embracing a set of common tasks observed in studies that select and integrate sources on-the-fly. As result from the taxonomy, we demonstrate how Built-up Integration features can be found in the literature, through an exemplification with related studies. We also highlight research opportunities regarding Built-up Integration, as a way to guide future development in a subdomain of Data Integration.

Keywords: Data Integration, On-the-fly Integration, Taxonomy

1 Introduction

The Big Data era raised the need to deal with large and heterogeneous data volumes on the Web and capture useful information for the end user [Oussous *et al*., 2018]. To this end, typical solutions involve information retrieval and natural language processing methods, and currently, approaches focused on the user experience started to gain more attention, in which the captured information is used as valuable knowledge to satisfy very specific needs [Frommholz *et al*., 2020]. Obtaining insights and actionable knowledge to meet specific requirements is not a trivial task, as it usually requires cross-analysis of data coming from multiple sources [Jovanovic *et al*., 2021]. In other words, satisfying user's needs often demands some form of *data integration* [El-Roby, 2018].

Data integration aims to join data from different data sources and provide users with a unified view [Li, 2017]. Historically, it involves different ETL (Extract-Transform-Load) flows regularly executed for augmenting data repositories such as Data Warehouses with external information [Jovanovic *et al*., 2016]. However, when we are dealing with highly specific and dynamic user needs, a periodical integration may be no longer viable, since the user expects quick solutions. Also, there might be cases in which the data at hand is insufficient to perform useful analyzes. In such cases, the integration should be performed *on-the-fly*, i.e., with data discovered, extracted, and merged *at query time*, so the information can become actionable [Nargesian *et al*., 2019]. In other words, assuming data that is scattered over multiple sources, on-the-fly integration makes it possible to provide the right information, at the right time [Nicklas *et al*., 2017].

However, there is a terminology mismatch in the literature involving on-the-fly integration: core activities linked to the concept can be found in the literature *under many different names*. E.g., when data integration is motivated by a specific context or situation, it is often called *situational data integration* (SDI), so data is integrated on-the-fly to deal with ad-hoc requirements and provide decision support [Han *et al*., 2013; Castellanos *et al*., 2012; Vo *et al*., 2018]. In other cases, the integration targets a cost-effective initialization, i.e., data is integrated as needed, postponing the addition of labor-intensive data, so it is called *pay-as-you-go integration* [Curry *et al*., 2019; Azuan, 2021]. Following this principle, the system incrementally understands and integrates the data over time by asking users to confirm matches on-the-fly, i.e., as the system runs [Jeffery *et al*., 2008]. On-the-fly integration also appears in the literature connected with *mashup applications*, which integrate data on-the-fly to provide a unique service for addressing immediate need in near real-time [Sehar *et al*., 2022; Liu *et al*., 2015; Missier *et al*., 2009; Daniel *et al*., 2018]. Moreover, the idea of discovering and joining data on-the-fly is often seen in the so-called *traversal-based approaches*, where the information is obtained by looking up data links related to a query, and intertwining them with the query result construction [Hartig and Freytag, 2012; Umbrich *et al*., 2015; Hartig and Özsu, 2016]. There is no unified nomenclature for embracing these concepts and other similar ones. Although each has its own particularities, they address highly connected (or even the same) activities: selecting and integrating data sources at query time, based on specific requirements, aiming to deliver good results to the end user.

Possibly the biggest issue concerning this lack of standardization is the disorientation it may cause to researchers of the area. E.g., if a researcher is interested in investigating one concept and is not aware of very similar concepts, his/her work could be significantly impacted. Furthermore, if dif-

ferent nomenclatures are frequently assigned to a common set of goals and tasks, eventually we will obtain numerous branches related to the same concept, hampering the search for features and their analysis in a comprehensive way. As a consequence, the development in this domain may be impaired. Improving knowledge organization in this area is, therefore, desirable.

This paper aims to bridge this gap and regulate knowledge in the area, by proposing the term **Built-up Integration**, which we define as *an approach that selects and integrates sources on-the-fly, supporting the user through augmented data.* The proposition of this term aims to help in the comprehension of similar concepts available in the literature, and embrace several data integration approaches that might be confused with each other. The proposed terminology is not a solution, but an organization proposition for integration studies that share tasks or methods.

Besides the concept definition, we based on related studies to propose a *taxonomy for Built-up Integration*, exposing main features that allow its identification, such as *Data Retrieval, On-the-fly Integration*, and *Data Delivery*. As result from the taxonomy, we demonstrate how Built-up Integration features can be found in the literature, taking as examples approaches that address traversal-based integration, SDI, data mashups, and pay-as-you-go integration. It is important to clarify that the present work does not aim to survey generic data integration approaches, since several studies already address these aspects [Halevy *et al*., 2006a; Cheatham and Pesquita, 2017; Halevy *et al*., 2006b; Ziegler and Dittrich, 2007]. In contrast, we present a taxonomy that can be used to organize and classify similar approaches that perform in an on-demand way.

The structure of the paper is presented as follows. In Section 2, we present a set of data integration concepts and discuss how they share similar tasks under different nomenclatures, demanding some kind of reorganization. In Section 3, we present a new terminology, named Built-up Integration, along with a taxonomy that unifies common features found in related studies. In Section 4, we discuss how Built-up Integration can be found in the literature, also discussing possibilities for the taxonomy extension. In Section 5 we discuss a set of opportunities for future researches. We conclude with our remarks in Section 6.

2 Overview of Data Integration Variants

Selecting and integrating sources on-the-fly are tasks addressed in many different studies in the literature. This section presents an overview of four data integration concepts that cover theses tasks, such as *mashups, pay-as-you-go integration, traversal-based integration*, and *Situational Data Integration*. Besides coping with data augmented in an on-demand way, these concepts were chosen as they present other correlated tasks regarding data sources selection and user support. Based on this existing connection, the current section also argues for the need to organize knowledge in the area through a new terminology.

2.1 Situational Data Integration

Situational Data Integration (SDI) is an integration that uses data sources discovered on-the-fly for dealing with specific and immediate queries and their dynamic requirements [Han *et al*., 2013; Wang *et al*., 2013]. The term "*Situational*" in SDI comes from the concept *Situation-Awareness* (SA), which is the perception of events related to an entity (i.e., the user) and the understanding of what is going on around, allowing to make accurate decisions [Kantorovitch *et al*., 2017]. By following this concept, a Situational Data Integration may be understood as an integration oriented by *situations of interest* to the user.

Although SA is a long-established concept [Endsley, 1995], situational integration only gained focus years later, in the *Business Intelligence* (BI) domain, where SDI was initially investigated due to its impact on operational decisions [Löser *et al*., 2008]. It came as an alternative to the traditional integration settings[1], aiming to analyze and combine large data sets (comprising both structured and unstructured data) for dealing with dynamic requirements and data-intensive flows [Löser *et al*., 2008; Jovanovic *et al*., 2016].

To clarify the SDI features, suppose that a retail company wants to analyze the success of its promotional campaign. The local data owned by the company (comprising products, customers, and orders data, for example) are not sufficient to perform such analysis and, more importantly, it is impracticable to wait for recent sales data to be periodically loaded into the Data Warehouse, since the company would like to *react faster* for improving the potential revenue. Then the business manager realizes that she can analyze campaign success in advance, by exploring the opinions or reviews that customers leave in the Web about product items. Therefore, she queries a system that will discover relevant data sources that contain the information needed, fetch data that relates to company's data and integrate this situational data with the local data [Abelló *et al*., 2013]. The integration results are then presented to the business manager, so she can improve business decisions.

In the scenario above, the local data are *stationary data*, and costumers' opinions are *situational data*, since they are not owned by the company and they have the role of providing a complete answer to a specific problem or need. Thus, when dynamic data is properly integrated, fused and correlated with stationary data, a situational picture can be derived, which is at the basis of the decision maker activities [Bonura *et al*., 2017].

Over the years, the term "Situational Data Integration" has been deprecated in the literature. The concept culminated around the last decade, and gradually, other terminologies started to be applied to talk about integration in situational scenarios [Han *et al*., 2013; Chen *et al*., 2017]. One example is the study in [Wang *et al*., 2013], where the authors develop a data mashup process that allows the recommendation of operators for performing situational integration. Following this discussion, data Mashups are addressed next.

[1] Based on the survey presented in [Jovanovic *et al*., 2016], we refer to "traditional integration settings" as those where data sources and query requirements are static, demanding a regular update of the data repository rather than real-time data flows.

2.2 Data Mashups

Data Mashups are usually Web-based applications that integrate data from multiple and heterogeneous sources, in order to provide a unique service [Paredes-Valverde *et al*., 2015]. They reuse and combine data sources (encapsulated as data services) on the Web, being developed in a rapid and ad-hoc manner to automate processes and mix information [Grammel and Storey, 2010]. Unlike applications aimed at expert users, mashups aim to move control over data closer to regular users, allowing to create applications through the merge of several existing data sources [Tran *et al*., 2014]. In website mashups, for example, the web page can be changed by removing elements, adding additional widgets, and changing their appearances [Grammel and Storey, 2010].

Mashups are not restricted to web applications, but they also support the development of situational applications for providing solutions to specific problems [Paredes-Valverde *et al*., 2015]. IoT-based mashups, e.g., support on-the-fly integration of contextual information such as real-time sensory data and historical data, so the decision-making process can be executed based on the integration [Cheng *et al*., 2018]. Another example comes from in [Chen *et al*., 2017]: the authors describe a highway emergency scenario, where the emergency staff should transfer wounded people to different kinds of hospital according to the injuries. In this case, the dispatcher needs to know estimated time of each ambulance to the target hospital to implement the proper scheduling. Based on this motivating scenario, the study proposes an approach for end users to discover data services and arrange them in a logical order, to finally create data service mashup plans automatically. Data service discovery and selection are performed among several candidate data services, e.g., *getInjuredInfo, getPersonInfo, getHospitalInfo*, and so on.

The source selection from the above example also allows us to highlight the dynamic and auxiliary nature of mashups when interconnecting several data sources: the user can control the services integration, in a way that he can use any service he wants, putting away the ones that he does not use anymore [Latih *et al*., 2011]. With respect to finding sources in mashups, users can mostly perform text-based searches, although context-specific suggestions can provide the needed elements to the user without requiring him to search [Grammel and Storey, 2010]. Indeed, many approaches in related literature focus on discovering and providing services with minimal manual settings [Sehar *et al*., 2022]. We can mention, e.g., the studies in [Lee and Kim, 2012; Lee, 2014], which propose algorithms to automate the discovery and composition of Web APIs, and the situational mashup in [Huang *et al*., 2008], in which the user context such as location and schedule determines the configuration of accessible widgets.

2.3 Traversal-based Integration

Traversal-based integration is a kind of virtual integration[2] that executes queries over Linked Data, traversing data links and merging up-to-date information from initially unknown sources [Hartig and Özsu, 2016; Mountantonakis and Tzitzikas, 2019]. The exploration of data links is performed at query execution time: first, it searches for URIs (Uniform Resource Identifiers) informed in the query body or as additional parameters. Secondly, it searches for more URIs that can possibly *enrich* the query results. The most relevant URIs and datasets for answering the query are selected, and finally, the answers from the sources are combined to return a final answer [Mountantonakis and Tzitzikas, 2019].

An example of traversal-based integration is given in [Hartig, 2014]: the authors consider a SPARQL query that asks for people who authored a paper about ontology engineering at some conference. This query cannot be answered from a single dataset, but requires data from the conference corpus, the names of the paper topics and the authors names. Thus, the traversal based query execution starts with some data retrieved from the conference corpus, by dereferencing the URI that identifies the proceedings. This data contains a set of RDF triples that match one of the triple patterns of the query, and results in Linked Data about published papers, including their topics. In the newly retrieved data, the query engine finds matching triples for the *publication* binding, so that solution mappings can be augmented with bindings for *topic*. Since the topics are also denoted by URIs, additional data can be retrieved to generate bindings for the topic label (e.g., ontology engineering). Following this strategy, it is possible to determine mappings that cover the whole query pattern and get to an integrated solution.

As well as situational integration and data mashups, traversal-based approaches offer up-to-date results, which are at the basis of user support [Umbrich *et al*., 2015]. That is because during the query execution, query links are traversed to expand the set of data already discovered, so further augmentations can be computed for partial solutions [Hartig and Freytag, 2012]. Concerning this expansion, the interesting part is that query execution (i.e., the link traversal) can start without a prior knowledge of available data sources. This *zero-knowledge* method is in line with the dynamic nature of the Web, motivating decentralization and dispensing with the use of data providers to setup costly endpoints [Fafalios and Tzitzikas, 2019].

Many traversal-based approaches for integration can be found in the literature. The authors in [Masmoudi *et al*., 2021] present a knowledge hypergraph-based approach, able to virtually integrate heterogeneous data from multiple sources and enhance the query answering process in terms of completeness. The SQUIN system [Hartig, 2013] discovers relevant data sources within RDF triples during the query execution, integrating the traversal of data links into the result construction. The system may be used either as a Java library that can be integrated in Web applications or as a Web interface. The proposal in [Harth *et al*., 2013] also supports on-the-fly integration, specifying the traversal method in rules. The approach sends requests to specific URIs, discovering links from where new data can be retrieved. Also, a software interface allows to poll the current state of resources at specific time intervals and react to updates, easing the transition from static to dynamic sources.

[2] A virtual integration assumes virtual repositories and the need for near real time data [Jarke and Quix, 2022].

Table 1. Comparison between different types of integration

Approach	Dominant Features	Input query	Type of Support	Human Involvement	Keywords
Situational Data Integration	Source discovery, Data augmentation, User support	Necessarily situational, various formats	Fresh data for decision-making	Mostly feedback	Business Intelligence, decision-making
Traversal-based Integration	Source discovery, Data augmentation	Mostly conjunctive queries	Data augmentation	Not required	URI exploration, zero-knowledge execution
Data Mashup	Source selection, Data augmentation, Human involvement	Mostly situational and GUI-based	Single service building, interactive interface	System operation (services composition)	User experience, control of the integration
Pay-as-you-go Integration	Data augmentation, Human involvement	Mostly conjunctive queries	Fresh data for decision-making, dynamic adaptation	Feedback and training	Low cost execution, User Feedback, Gradual improvement

2.4 Pay-as-you-go Integration (Dataspaces)

Providing a coherent view of data is a classical challenge for data integration: although automatic approaches can bring together lots of correct, valuable information, they also may present a fair amount of misleading data [Paton *et al*., 2012]. Another classical challenge concerns the high cost of integration initialization, which demands the automatic inference of schema matches and semantic mappings [Maskat, 2016]. These tasks are able to produce highly accurate results, but usually involve a delayed start-up time.

To overcome these drawbacks and provide a cost-effective data integration, the variant *pay-as-you-go integration* was proposed, also known as *dataspaces* [Halevy *et al*., 2006a]. The idea of this variant is to distribute the costs of data integration creation to other stages of the integration process, by starting the initialization of dataspaces at the earliest opportunity, and also gathering feedback from the user to improve the integration [Azuan, 2021]. In this approach, the assumption is that some application contexts do not require full integration in order to provide useful services, so data is integrated on an "as-needed" basis, with the labor-intensive aspects of data integration postponed until they are required, and when tighter semantic integration is required, it can be achieved in an incremental "pay-as-you-go" way [Curry *et al*., 2019; Das Sarma *et al*., 2008]. The user feedback is also gathered in a continuously manner, throughout the entire lifespan of the dataspace, so a better quality in the integration is achieved with lower upfront-cost.

As well as other integration approaches such as SDI or mashups, pay-as-you-go integration aims to support the user by meeting his requirements. Hence, it is necessary to identify and select data sources that can effectively provide complete answers or results [Azuan, 2021]. The pay-as-you-go integration is specially well suited with unstable query requirements and sources that may change rapidly, as it consumes data at an on-demand recombination perspective [Furche *et al*., 2016]. Consider, e.g., an e-commerce company that wants to compare price among competitors; relevant sources come and go frequently, and both format and contents change regularly. A classical integration that produces perfect results would not be practical nor effective to integrate the relevant sources and support well-informed decisions [Paton *et al*., 2016].

The literature shows several studies that address pay-as-you-go integration and data management. E.g., the study in [Herzig and Tran, 2012] proposes to query data using on-the-fly mappings, which support a pay-as-you-go paradigm where data is embedded into the search process. Also, the authors in [Serrano *et al*., 2018] propose to quantify the quality of an integration: given a set of mappings and a set of workers of unknown trustworthiness, feedback instances are collected in the extents of the mappings that characterize the integration.

2.5 Compiling Concepts

As observed in the previous subsections, some data integration approaches share many functions and goals, slightly differing from each other as some have a greater focus on one task than others. This is shown on Table 1, which summarizes characteristics that best differentiate the integration approaches, such as the type of input query, type of user support, and the level of human involvement. These characteristics are mentioned in the related papers discussed in the previous subsections. Besides these, the comparative table also shows dominant features (i.e., the main focus of each integration approach) and keywords (for which goal they are usually applied). We can see that some concepts are more focused on the data discovery process (e.g., traversal-based approaches) or the decision-making support (e.g., SDI). Also, in some of them (such as Data Mashups and Pay-as-you-go integration), the user role is more significant.

But most importantly, with regard to the *similarities* between the approaches, we recognized that, in general, they perform source discovery and/or selection, some kind of data augmentation made at query time, and have the common goal of supporting the end user (either by helping him to make a decision, or by providing valuable visualization from the integrated data). All of them may involve multiple and heterogeneous data sources in the integration process. In addition, it is possible to find in the literature studies that mention more than one concept, i.e., SDI and mashups [Wang *et al*., 2013; Cheng *et al*., 2018], mashups and pay-as-you-go inte-

gration Tatemura *et al*. [2008]; Franklin *et al*. [2008]; Hirmer and Mitschang [2017], or even traversal-based approaches and mashups [Hartig, 2013; Matskanis *et al*., 2012].

Despite the similarities found, there is still a lack of a nomenclature for unifying all associated tasks. The problem with this gap is the confusion it generates for researchers that may be interested on a concept often addressed in the literature under another name. Let us suppose, for example, that a researcher wants to systematically review all studies in the literature that address the term "Situational Data Integration" (SDI), to check how the discovery of data sources takes place in a certain period of time. In this case, several studies that present the same idea as SDI under a completely different name could be ignored, since the researcher is not aware of the similarities and differences among the existing concepts. As a result, this "non-awareness" would certainly impact on the reliability of the research produced. Thus, a taxonomy for properly organizing the knowledge in this area is needed.

In the next section we define a novel terminology named **Built-up integration**, aiming to regulate similar aspects observed in several integration approaches. We also present a taxonomy for Built-up integration, covering a set of features related to source selection and discovery, data integration and information delivery.

3 Reorganizing the Knowledge: Built-up Integration

Based on common features found in different integration methods (see Section 2), this section proposes a novel term named **Built-up Integration**, which follows the following definition:

> *Built-up Integration selects and manages data sources on-the-fly for dealing with specific query requirements, resulting in an augmented data set for supporting the user.*

The term is proposed to assign a common term to data management characteristics that are found together in several integration approaches, such as the ones mentioned in Section 2. In these approaches, the information value is often transient, so that applications consume data on-the-fly to perform specific and/or additional analysis tasks demanded by user requirements. We believe the term *Built-up Integration* is appropriate to accommodate such approaches, since it expresses the idea of data being combined gradually and systematically, until reaching a set of unified data, complete enough to support the end user. At this basis, a Built-up integration system must systematically analyze potential data sources at hand and select the one(s) that can meet the users needs. The selected sources must be reconciled and combined towards the delivery of up-to-dated solutions, which cannot be addressed by loading data repositories from time to time. The user has a central role in the integration, since data management occurs targeting a timely support and, desirably, a positive impact in his decisions. In addition, the user can participate actively in all tasks involved in Built-up integration, by deciding which information are relevant or giving feedback on the responses received.

Beyond a formal definition of Built-up integration, in this section we also propose a taxonomy that unifies a set of features that are found in related literature. The taxonomy (shown in Figure 1) was created as a feature diagram using the Feature-Oriented Domain Analysis method [Kang *et al*., 1990]. A feature diagram is a hierarchically arranged set of features, where relationships between a parent feature and its child features may be categorized as: and – all subfeatures must be selected, $alternative$ – only one subfeature can be selected, $inclusive\ or$ – one or more can be selected, $mandatory$ – features that are required, and $optional$ – features that are optional [Batory, 2005].

For building the taxonomy from related studies (see Section 2), we first searched for integration-based approaches on Google Scholar search engine[3], using keywords such as *situational integration, mashup systems, on-the-fly integration, linked data, traversal-based query, pay-as-you-go integration*, alternately and merging the keywords for filtering results. We also used the Connected Papers online tool[4] for verifying related and derivative papers.

Next, we extracted a set of characteristics from the related studies, grouping them by similarity. A generic nomenclature was assigned for each group, so that similar characteristics in a group were represented in an unified way. By following this process, the Built-up integration taxonomy was built, composed by three main features: **Data Retrieval**, **On-the-fly Integration**, and **Data Delivery**, which represent important steps executed by several recent data integration systems. These features are detailed as follows.

The first main feature, **Data Retrieval**, covers the ability of retrieving useful sources of information for dealing with specific domain problems. For doing this, it relies on two mandatory subfeatures, named `Input Query` and `Source Selection`. Through the information contained in the input query, a system can analyze different candidate sources of information and select the ones that are most likely to provide a reliable answer. A user query can be expressed through many ways: by using a particular language such as SQL or SPARQL (which are types of conjunctive queries), through natural language, by interacting with GUI-based elements in an interface, and so on.

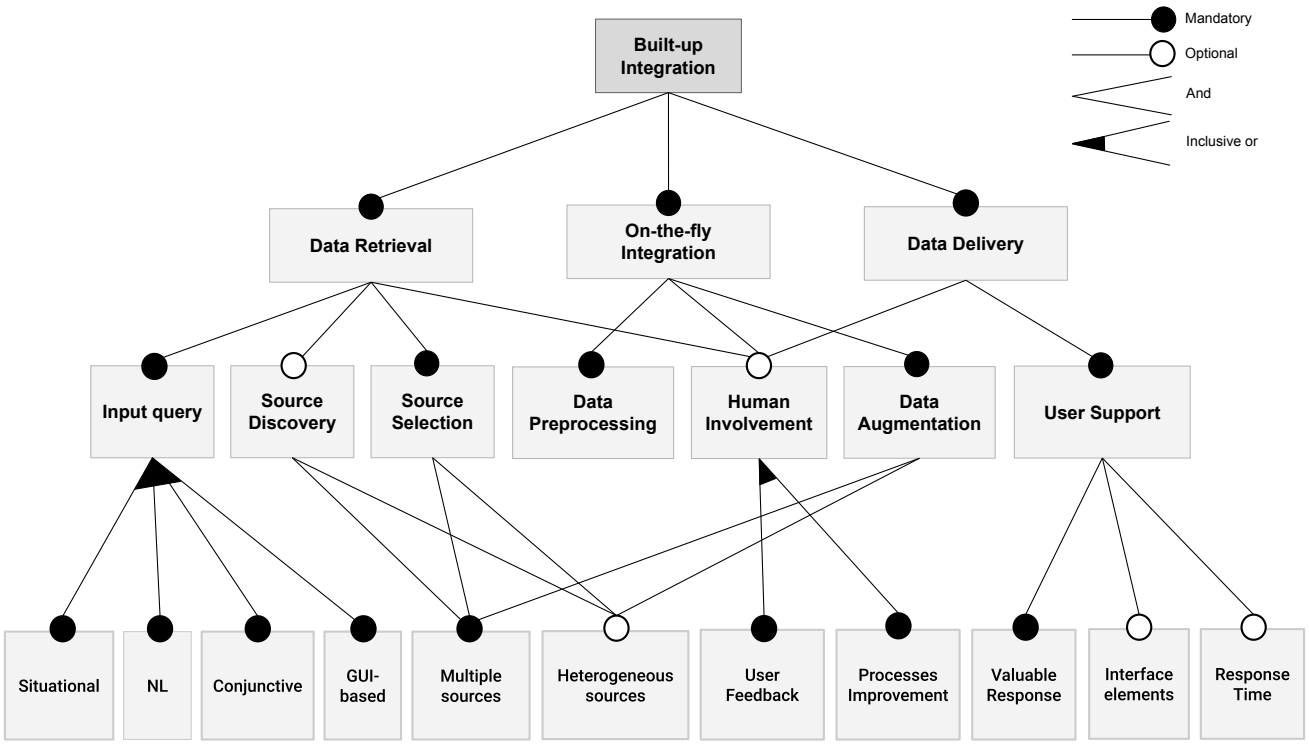
Based on an input query, the integration system can perform `Source Discovery`. This feature assumes that data sources must be discovered on-the-fly for dealing with users requirements [Abelló *et al*., 2014]. The discovery may be explored through table relatedness measures [Zhang and Ives, 2020], similarity joins between data collections [Xu *et al*., 2019], or general similarity methods that determine how related the source is according to the query. `Human Involvement` may be present during Data Retrieval, if the system allows the user to solve ambiguities in data sources attributes, or choose one source among several good matches.

As the second main feature of the taxonomy (see Figure 1), there is **On-the-fly Integration**. This feature refers to the ability of analyzing and combining, *at query time*, heterogeneous data from the previously discovered sources. Hence, it should also consider possible changes in users requests,

[3] https://scholar.google.com

[4] https://www.connectedpapers.com/

Figure 1. Taxonomy for Built-up Integration



adapting data sources and/or processes accordingly [Jovanovic *et al*., 2016]. Due to these characteristics, an integration performed on-the-fly allows to retrieve situational data, from which we can derive valuable insights that enable accurate decisions [Vo *et al*., 2018]. Regarding the taxonomy, On-the-fly Integration has `Data Preprocessing`, `Data Augmentation`, and `Human Involvement` as subfeatures.

`Data Preprocessing` is a step that aims to deal with heterogeneous data sources that usually contain out-of-date, noisy, or conflicting data. In order to perform a proper integration, the information extracted from a source needs to be accessible through data cleansing methods, which should resolve unique entities or fill up missing information [Löser *et al*., 2008]. Also, data integration often requires some sort of correlation, by means of obtaining the data schema from the discovered sources, estimating available dimensions, facts, and measurements, performing similarity calculation, and so on [Nargesian *et al*., 2019].

After cleaning the data and assessing how they can be used to meet the users needs, the selected data sources are finally merged, causing an "extension" of the information previously available. We call this feature `Data Augmentation`, which can occur by combining stationary data (e.g., a DW) with external data [Vo *et al*., 2018], or simply merging external resources [Hartig and Özsu, 2016]. The resulting augmented set is then used for answering requests that could not be properly answered, due to lack or insufficiency of the current data [Abelló *et al*., 2013].

During the integration, `Human Involvement` may be present to help the system to improve the quality of data integration. By collecting user feedback and using it as training tool, it is possible to minimize errors, improve matching tasks, and generate high-quality rules for a system operation [Li, 2017]. This participation is not only valuable during the data integration, but also in the Source Discovery step, where human knowledge can solve ambiguities and indicate relevant sets of data.

Data Delivery is the last feature of the Built-up Integration taxonomy, and it refers to the just-in-time delivery of valuable solutions to the user after completing the integration. Particularly, Data Delivery demands `User Support`, i.e., making the user aware of situations that can potentially affect his activities. This can be done by the system simply producing a valuable response (e.g., providing some explanation of how calculations took place), or making a recommendation (e.g., which sequence of steps the user should apply) [Serban *et al*., 2013]. This kind of support can be assisted by a graphical interface, favoring the user interaction and feedback collection. Also, another way to support the user is through methodologies for improving the system's response time by, e.g., reducing the processing complexity, implementing multi-thread execution, and so on [Löser *et al*., 2008]. Finally, as well as the other main features, Data Delivery can also have `Human Involvement`, where user feedback about the responses can refine the results and fine-tune data models.

Several approaches in the literature perform Built-up integration, since they have mechanisms to jointly execute Data Retrieval, On-the-fly integration, and Data Delivery (the main features of the taxonomy). The next subsection presents examples of them, indicating how Built-up integration can be found in data integration studies.

4 Where we can find Built-up Integration?

This section aims to correlate Built-up Integration with the types of data integration discussed in Section 2, in order to provide an overview of how the proposed features can be identified in the literature.

4.1 The Main Features

As mentioned in Section 3, three main features are often shared between integration approaches, which are *Data Retrieval, On-the-fly integration*, and *Data Delivery*. The Table 2 exemplifies how these features can be found, taking as a basis a set of related studies mentioned in the Section 2. Besides the table, the main features identification is discussed in details next.

4.1.1 Data Retrieval

As mentioned in Section 3, Data Retrieval requires `Input Queries` and `Source Selection`. Concerning the former, we assume that users needs may be expressed in many different ways. Conjunctive queries (such as SQL and SPARQL) are the most common class of queries used in database systems, and are used in traversal-based/pay-as-you-go approaches [Arenas *et al*., 2021; Hartig and Freytag, 2012; Hedeler *et al*., 2009]. Source Selection in mashup approaches often relies on a graphical interface, so the user can interact with components, and overload models until a satisfactory solution is found [Sehar *et al*., 2022]. Mashups can be also built with programming languages such as EMML (Enterprise Mashup Markup Language), Orc, and YQL (Yahoo Query Language) [Paredes-Valverde *et al*., 2015].

Regardless of query format, in SDI the input query is *situational*, i.e., it focuses on a particular problem and cannot be defined in advance [Abelló *et al*., 2013; Wang *et al*., 2013]. Mashup queries can also be situational, since a mashup is usually developed for rapidly address an immediate need or a specific situation [Latih *et al*., 2011]. In pay-as-you-go integration, the querying service can be made through keyword search, structured queries (assuming that the user understands the underlying data schema), browsing of available datasets, and even through question-answering (which focus on natural language interaction) [Curry *et al*., 2019].

The most challenging task related to source selection in Built-up integration is `Source Discovery`. In some cases, users spend more time searching for relevant information than analyzing it, and for facing this issue, Source Discovery automatically finds one or more data sources suitable to a user query. This taxonomy feature can be found in several integration systems.

In SDI, a common assumption is the existence of a local database, although it usually cannot provide an actionable information to the user by itself. Thus, a source discovery engine should discover external information to be further integrated with the local data [Vo *et al*., 2018]. The data sources are determined according to particular query requirements, or to previous integration results [Wang *et al*., 2013; Löser *et al*., 2008]. Regarding traversal-based integration, existing approaches can discover initially unknown data sources at runtime, so they start querying without first having to populate a repository of data [Hartig and Özsu, 2016]. In data mashups, services can be discovered by, e.g., text-based searches and browsing of services' structural properties, whereas the content is mostly collected with the help of APIs (Application Programming Interfaces) [Grammel and Storey, 2010; Sehar *et al*., 2022].

In pay-as-you-go approaches, new sources are included in the dataspace automatically. In this kind of integration, the semantic relationships derived may be approximate, but the inclusion of user feedback (`Human Involvement`) assists in gathering information about the selected sources and dealing with data uncertainty [Azuan, 2021]. In fact, `Human Involvement` in Data Retrieval can also be observed in SDI approaches, because when situational data are retrieved and returned, the user may decide that they are not suitable for the task at hand [Abelló *et al*., 2013].

4.1.2 On-the-fly Integration

On-the-fly integration combines sources at query time, aiming to satisfy a situational need (i.e., a specific and ad-hoc requirement). This feature covers `Data Preprocessing` and `Data Augmentation` as important subfeatures within Built-up integration, and it can be observed in derived approaches.

In SDI, the integration joins situational/external data with an information previously available, generating an augmented set of data, which is used to provide useful insights [Vo *et al*., 2018]. The situational approaches described in [Jovanovic *et al*., 2021; Nadal *et al*., 2019; Ferrández *et al*., 2016] exemplify data preprocessing and augmentation covered by Built-up integration. Pay-as-you-go integration implies that resource-intensive data integration should be performed at much lower cost, thereby it occurs on demand, starting with a lower data quality. As a result, the approaches make use of techniques that infer relationships between resources and refine these relationships in the light of user feedback [Hedeler *et al*., 2009]. In relation to Data Augmentation, pay-as-you-go approaches execute a *data fusion* step, where the dataspace instances are transformed into a single and consistent representation instance, which will be later available for user viewing [Maskat, 2016].

Proposals that cover traversal-based integration also present some sort of Data Augmentation. In link traversal, for example, data links may be traversed during the query execution to expand discovered data, i.e., to augment a dataset [Hartig and Freytag, 2012]. Such augmentations can also be found in graph-based approaches that execute traversal algorithms for integration [Kordjamshidi *et al*., 2017; Qi and Luo, 2016]. With respect to Mashups, data from different sources are merged into a single joint place. The result from this augmentation can be visualized as a web page, a web application, or a service, which is able to fulfill users requirements [Sehar *et al*., 2022].

Table 2. Built-up Integration examples in the literature

Approach	Type of Integration	Data Retrieval Features	On-the-fly Integration Features	Data Delivery Features
[Hartig and Freytag, 2012]	Traversal-based	Conjunctive queries, discovery and selection of URIs.	Link exploration made at query time, the discovered URIs are used to enrich the results.	Provide a complete answer to a question, achieved with data augmentation.
[Umbrich *et al*., 2015]	Traversal-based	Conjunctive queries, prototype with a source selector that decides which query and URIs should be dereferenced, and which links should be followed.	Query evaluation strategy that discovers additional sources on-the-fly and integrate data during query-answering.	Offers the potential to get fresh answers when dynamic information is involved.
[Ferrández *et al*., 2016]	SDI	Situational and NL query, selection of external sources (obtained with a Question-Answering system) to be integrated in a Data Warehouse system.	Full integration of unstructured and structured information, allowing to compare data instantaneously through a dashboard.	Data delivery allows the user to take quick strategic decisions based on richer data.
[Nadal *et al*., 2019]	SDI	Situational and conjunctive queries. Situational data is achieved by means of RESTful APIs.	The proposed Big Data Integration ontology semi-automatically adapts to situational data acquired under a schema evolution process.	The proposal aims to evolve decision making and exploits end-user feedback to improve the quality of experience.
[Wang *et al*., 2013]	SDI/mashup	Situational and GUI-based queries, selection of several data sources by means of web services.	Retrieved data can be accessed and combined, and finally published as a composite data service.	The user is involved and supported in the data mashup process, as the system provides interactive recommendation of composition operators for performing situational integration.
[Curry *et al*., 2019]	Pay-as-you-go	Conjunctive queries, dataspace query service for real-time data streams that enables unified queries across live streams, historical data, and entities.	Real-time query service which preprocesses the streams on-the-fly instead of storing them, complementing older views already achieved.	Validation of the Real-time Linked Dataspace proposed within five real-world smart environments pilot deployments to build real-time analytics, decisions support, and smart apps for smart energy and water management.
[El-Roby, 2018]	Pay-as-you-go	Conjunctive queries, interface-based discovery and selection of RDF data sets. Human feedback collected during the interaction is used to reject links and discover new links similar to the ones approved by the user.	Discovered links are automatically incorporated in the query processing, aiming to achieve a complete output/answer.	Data delivery provides more complete answers to the user.
[Cheng *et al*., 2018]	Mashup/SDI	GUI-based queries to select the different types of data sources, such as sensory data, local files, relational databases, and Web service resources.	The situational IoT services mashup approach supports on-the-fly integration of the different data services.	Decision-making processes can be executed based on real-time sensed data. The user is also supported in the composition of situational applications.
[Lee and Kim, 2012]	Mashup	GUI-based queries, automatic discovery and selection of Web APIs.	Graph-based composition algorithm for the integration of Web APIs, where a composition is gradually generated by a backward chaining of APIs. At each step, suitable APIs are automatically added to the composition.	Users can obtain immediate composition results visually, and iteratively refine their goals to achieve improved results.
[Tatemura *et al*., 2008]	Mashup/Pay-as-you-go	GUI-based queries, source discovery and selection based on automated schema matching. The system lets the user refine the results interactively.	The system helps a user to improve the results from sources integration, at a query time, until he is satisfied (along with the "pay-as-you-go" principle).	Query results can be visualized in an interface, where the user can also give feedbacks. If the result is satisfactory, it can be saved in several ways, dynamic spreadsheets or a new web service.
[Hartig, 2013]	Traversal-based/Mashup	Conjunctive queries. The SQUIN proposal discovers and retrieves data that might be relevant for answering a query during the query execution process itself. A mashup application queries the Web of Linked Data using SQUIN.	Incremental construction of query results with the traversal of data links. Suitable for an "on-demand" live querying scenario.	Data delivery of fresh answers.

4.1.3 Data Delivery

After data integration, the user needs to receive and visualize problem solution in an effective way. In the taxonomy, this feature is called *Data Delivery*, which covers `User Support` and `Human Involvement`.

User support is a key feature for SDI: as it bases on Situation-Awareness, the integration focuses on providing decision-making in complex and dynamic situations. The support can be achieved by means of predictions, alerts, or recommendations given to the user [Bonura *et al*., 2017]. The general idea is that data integration can make users aware of the current situation, and hence they have the opportunity to take immediate action [Castellanos *et al*., 2012]. Pay-as-you-go integration also focuses on supporting decision-making in highly dynamic environments. In this setting, a practical and fast integration is better than a perfect integration, since the user can revise the results and gradually improve the process [Maskat, 2016].

In Data Mashups, user support occurs mostly through an effective visualization of results, i.e., graphical interfaces that allow the user to combine services and see solutions that meet the initial requirements [Sehar *et al*., 2022]. User support can also mean an improvement of user experience. Traversal-based integration, for instance, covers query optimization techniques and settings for URI lookups that aim to reduce the response time [Hartig and Özsu, 2016].

In the proposed taxonomy of Built-up Integration, Human Involvement is not mandatory. Some integration approaches (such as the traversal-based ones) do not require user guidance or feedback, whereas for others, this kind of human participation is essential. E.g., in pay-as-you-go approaches, human feedback is highly important to indicate the correctness of the received answers, constantly improving the integration [El-Roby, 2018]. Similarly, in SDI, fused data may be approved by the user, who can either confirm the results or propose alternatives [Abelló *et al*., 2013]. Also, data mashups can be executed in a semi-automated way, with user guidance during data discovery and integration [Paredes-Valverde *et al*., 2015].

4.2 An Embracing Taxonomy

We proposed Built-up Integration for systematically grouping and labeling similar tasks found in integration-based approaches. It is important to recall that the term does not come to fully replace the terms already defined in the literature, since each one has its particularities (see Table 1), especially considering the research context in which they were defined. In this sense, it would be correct to affirm, e.g., that SDI is a type of Built-up Integration (as it contains the characteristics defined in the proposed taxonomy), but keeping distinct characteristics, such as its application in strategic decision processes and the premise of stationary data. Most importantly, the Built-up Integration terminology highlights the similarities between the approaches *already available* in the literature, and makes way to *future* approaches to be better categorized.

For demonstration purposes, the present section only analyzed Built-up Integration related to four integration concepts existing in the literature, which were chosen due to the many common tasks identified, and whose connection can be even more strengthened if we consider the joint mentions in past publications (see Table 2). However, besides these concepts, other types of integration executed on-demand (or considering situational problems) could also be considered and classified within the taxonomy features. The taxonomy is generic, meaning that it can be extended in the future to cover more detailed aspects such as data sources types, adaptation methods for source selection, or preprocessing techniques. In this case, the level of detail should consider the need to specify optional edges, i.e., a feature optional on one side and mandatory on another [Schobbens *et al*., 2006].

Regardless of the taxonomy granularity, we consider that Built-up Integration features are still very challenging. When observing characteristics such as Input Query and Source Discovery, for example, we identify different efforts in the literature for providing the user the best experience as possible, either by focusing on natural language or interactivity. At this basis, the next section presents some research opportunities concerning Built-up Integration, as a way to motivate and provide guidance for future researchers.

5 Thinking Ahead: Research Opportunities for Built-up Integration

Nowadays, users can benefit from many tools (e.g., Hadoop or Apache Spark) that manage data governance, discovery, extraction, cleaning, and integration [Nargesian *et al*., 2019]. Although this helps creating and consuming information, many challenges still remain.

Considering that Built-up Integration represents an umbrella term for several kinds of integration, its challenges involve how to execute its steps (see Figure 1) at query time, considering the highly heterogeneous structure of datasets, and leveraging user experience. When it comes to situational requirements, existing applications face uncertainty when discovering data with automatic matchers [Liu *et al*., 2015]. Purely automated methods cannot fully address this issue, and beyond that, they infer the best way to integrate data sources based on the features of these sources, which can output several errors [El-Roby, 2018]. As a consequence, data analysis and user support is affected.

One way to minimize errors and uncertainties in Built-up Integration when discovering data sources is to integrate human knowledge for solving ambiguities, improving matching tasks, and even generating high-quality rules for a system operation [Li, 2017]. In cases where the integration system makes a recommendation, the user feedback can help refine the results and fine-tune data models. Although beneficial, this human inclusion also requires considering *how to efficiently collect the user knowledge.* Indeed, there is a lack of unified platforms accessible to users without technical skills [Jovanovic *et al*., 2021; Khalajzadeh *et al*., 2018]. Many integration approaches that refine their processes based on user feedback expose problems through technical details and ask the user to fix them, implicitly assuming that the user is an expert [El-Roby, 2018].

We raise the hypothesis that conversational interfaces such

as *Question-Answering* (QA) systems can be applied to deal with these issues. By using a QA system as interaction and mediation tool, human knowledge can be detected and used for improving many tasks within the system operation [Li, 2017]. As a consequence, Built-up Integration tasks such as source discovery and on-the-fly integration could become more efficient. Also, besides capturing user feedback, QA systems are user-friendly solutions to perform situational tasks (e.g., accessing multiple and heterogeneous sources, combining data, visualizing integration results), which are usually restricted to experienced computer users [Paredes-Valverde *et al*., 2015]. They act as an access point to data sources, obtaining fast results and delivering them in natural language [Daniel *et al*., 2020].

Metadata discovery is also an interesting direction to follow towards Built-up Integration, since it favors data understanding and the identification of relevant data to be integrated [Nargesian *et al*., 2019]. The opportunities come from the fact that sources may be modeled and structured in many ways, e.g., with attributes named differently in data schemas, thus demanding the effective exploration of semantics and mapping methods [Hai *et al*., 2020]. Built-up Integration also requires to consider the lack of necessary metadata for source discovery, so methods for schema inference can be valuable for metadata enrichment [Koupil *et al*., 2022].

Finally, decision-making solutions have been continuously explored in the last years [Duan *et al*., 2019], representing a promising opportunity for Built-up Integration's Data Delivery. It is desirable that future research can go beyond delivering answers in an agile time, by favoring the taking of strategic actions. In other words, discovering and merging sources at query time through Built-up Integration are steps to be explored with a view to improving services and enabling smarter decisions. E.g., in Business Intelligence (BI), integrated information can be used for obtaining competitive advantages and leverage organizational collaboration [Jovanovic *et al*., 2016], while in Open Data, it can favor governmental transparency [Miller, 2018]. Reinforcement learning algorithms can be investigated for these purposes, as they help finding optimal strategies in terms of prediction, being extensively applied for Big Data processing [Singh *et al*., 2022; Derakhshan *et al*., 2019].

6 Conclusions

This paper presented *Built-up Integration*, a term for embracing common features encountered in several studies that retrieve and merge data sources at query time. We identified a lack of consistency regarding the terminologies in the area, since tasks related to *data retrieval, on-the-fly integration*, and *data delivery*, despite being often used together, are recognized in the literature under different names. The lack of a common terminology in this context not only makes it difficult to find and assess a group of tasks or concepts, but also gives space for more and more different names to be assigned to similar methodologies.

For regulating knowledge in the area, Built-up Integration was proposed as a type of integration where data sources are selected and managed on-the-fly, towards user support. Beyond a formal definition, we proposed a taxonomy that organizes similar characteristics found in related literature through features and subfeatures, which follow a unified nomenclature. We also correlated Built-up Integration with other data integration variants (such as situational data integration, mashups, traversal-based integration, and pay-as-you-go integration), which are rarely analyzed together, to exemplify where the taxonomy features can be found in existing approaches. By highlighting intersections among different types of integration, the proposed taxonomy has potential to organize current and future knowledge produced, allowing the researchers to classify as Built-up Integration those approaches that execute Data Retrieval, On-the-fly Integration, and Data Delivery tasks.

In terms of future development, we discussed some research opportunities for Built-up Integration, such as the efficient use of human feedback for dealing with uncertainties, the use of conversational interfaces as mediation tools, as well as metadata discovery and reinforcement learning methods for improving user support. Regarding these opportunities, our future work involve the use of Built-up Integration features in a Question-Answering system architecture, as a way to exploit human feedback in situational contexts.

Finally, as the taxonomy is an organization proposition for similar tasks and methodologies shared among studies, it is generic and can be extended to cover more detailed aspects on source retrieval and integration performed at query time. This means that other knowledge management contributions can be derived from the taxonomy, such as the comparison among methods used in each group of features, or the analysis of methods within Built-up Integration that also present a lack of nomenclature standard.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

References

Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J.-N., Naumann, F., Pedersen, T., Rizzi, S. B., Trujillo, J., Vassiliadis, P., *et al*. (2013). Fusion cubes: Towards self-service business intelligence. *International Journal of Data Warehousing and Mining (IJDWM)*, 9(2):66–88.

Abelló, A., Romero, O., Pedersen, T. B., Berlanga, R., Nebot, V., Aramburu, M. J., and Simitsis, A. (2014). Using semantic web technologies for exploratory OLAP: a survey. *IEEE transactions on knowledge and data engineering*, 27(2):571–588.

Arenas, M., Croquevielle, L. A., Jayaram, R., and Riveros, C. (2021). When is approximate counting for conjunctive queries tractable? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1015–1027.

Azuan, N. A. A. (2021). *Exploring Manual Correction as a Source of User Feedback in Pay-As-You-Go Integration*.

PhD thesis, The University of Manchester (United Kingdom).

Batory, D. (2005). Feature models, grammars, and propositional formulas. In *International Conference on Software Product Lines*, pages 7–20. Springer.

Bonura, S., Cammarata, G., Finazzo, R., Francaviglia, G., and Morreale, V. (2017). A novel webGIS-based situational awareness platform for trustworthy big data integration and analytics in mobility context. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 86–98. Springer.

Castellanos, M., Gupta, C., Wang, S., Dayal, U., and Durazo, M. (2012). A platform for situational awareness in operational BI. *Decision Support Systems*, 52(4):869–883.

Cheatham, M. and Pesquita, C. (2017). Semantic data integration. *Handbook of big data technologies*, pages 263–305.

Chen, X., Han, Y., Wen, Y., Zhang, F., and Liu, W. (2017). A keyword-driven data service mashup plan generation approach for ad-hoc data query. In *2017 IEEE International Conference on Services Computing (SCC)*, pages 394–401. IEEE.

Cheng, B., Zhao, S., Qian, J., Zhai, Z., and Chen, J. (2018). Lightweight service mashup middleware with REST style architecture for iot applications. *IEEE Transactions on Network and Service Management*, 15(3):1063–1075.

Curry, E., Derguech, W., Hasan, S., Kouroupetroglou, C., and ul Hassan, U. (2019). A real-time linked dataspace for the internet of things: enabling "pay-as-you-go" data management in smart environments. *Future Generation Computer Systems*, 90:405–422.

Daniel, F., Matera, M., Quintarelli, E., Tanca, L., and Zaccaria, V. (2018). Context-aware access to heterogeneous resources through on-the-fly mashups. In *International Conference on Advanced Information Systems Engineering*, pages 119–134. Springer.

Daniel, G., Cabot, J., Deruelle, L., and Derras, M. (2020). Xatkit: a multimodal low-code chatbot development framework. *IEEE Access*, 8:15332–15346.

Das Sarma, A., Dong, X., and Halevy, A. (2008). Bootstrapping pay-as-you-go data integration systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 861–874.

Derakhshan, B., Mahdiraji, A. R., Rabl, T., and Markl, V. (2019). Continuous deployment of machine learning pipelines. In *EDBT*, pages 397–408.

Duan, Y., Edwards, J. S., and Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda. *International journal of information management*, 48:63–71.

El-Roby, A. (2018). *Web Data Integration for Non-Expert Users*. PhD thesis, University of Waterloo.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human factors*, 37(1):32–64.

Fafalios, P. and Tzitzikas, Y. (2019). How many and what types of SPARQL queries can be answered through zero-knowledge link traversal? In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 2267–2274.

Ferrández, A., Maté, A., Peral, J., Trujillo, J., De Gregorio, E., and Aufaure, M.-A. (2016). A framework for enriching data warehouse analysis with question answering systems. *Journal of Intelligent Information Systems*, 46(1):61–82.

Franklin, M., Halevy, A., and Maier, D. (2008). A first tutorial on dataspaces. *Proceedings of the VLDB Endowment*, 1(2):1516–1517.

Frommholz, I., Liu, H., and Melucci, M. (2020). Birds-bridging the gap between information science, information retrieval and data science. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2455–2458.

Furche, T., Gottlob, G., Libkin, L., Orsi, G., and Paton, N. (2016). Data wrangling for big data: Challenges and opportunities. In *Advances in Database Technology—EDBT 2016: Proceedings of the 19th International Conference on Extending Database Technology*, pages 473–478.

Grammel, L. and Storey, M.-A. (2010). A survey of mashup development environments. In *The smart internet*, pages 137–151. Springer.

Hai, R., Miller, R., Jarke, M., and Quix, C. J. (2020). *Data Integration and Metadata Management in Data Lakes*. PhD thesis, Ph. D. Dissertation. RWTH Aachen University. DOI: https://doi. org/10.18154 ….

Halevy, A., Franklin, M., and Maier, D. (2006a). Principles of dataspace systems. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–9.

Halevy, A., Rajaraman, A., and Ordille, J. (2006b). Data integration: The teenage years. In *Proceedings of the 32nd international conference on Very large data bases*, pages 9–16.

Han, Y., Wang, G., Ji, G., and Zhang, P. (2013). Situational data integration with data services and nested table. *Service Oriented Computing and Applications*, 7(2):129–150.

Harth, A., Knoblock, C. A., Stadtmüller, S., Studer, R., and Szekely, P. (2013). On-the-fly integration of static and dynamic linked data. In *Proceedings of the Fourth International Workshop on Consuming Linked Data co-located with the 12th International Semantic Web Conference*, pages 1613–0073.

Hartig, O. (2013). SQUIN: a traversal based query execution system for the web of linked data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1081–1084.

Hartig, O. (2014). *Linked Data Query Processing Based on Link Traversal*, pages 263–283. DOI: 10.1201/b16859-15.

Hartig, O. and Freytag, J.-C. (2012). Foundations of traversal based query execution over linked data. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 43–52.

Hartig, O. and Özsu, M. T. (2016). Walking without a map: Ranking-based traversal for querying linked data. In *International Semantic Web Conference*, pages 305–324. Springer.

Hedeler, C., Belhajjame, K., Fernandes, A. A., Embury, S. M., and Paton, N. W. (2009). Dimensions of dataspaces. In *British National Conference on Databases*, pages 55–66. Springer.

Herzig, D. M. and Tran, T. (2012). Heterogeneous web data search using relevance-based on the fly data integration. In *Proceedings of the 21st international conference on World Wide Web*, pages 141–150.

Hirmer, P. and Mitschang, B. (2017). TOSCA4Mashups: enhanced method for on-demand data mashup provisioning. *Computer Science-Research and Development*, 32(3-4):291–300.

Huang, A. F., Huang, S. B., Lee, E. Y., and Yang, S. J. (2008). Improving end-user programming with situational mashups in web 2.0 environment. In *2008 IEEE International Symposium on Service-Oriented System Engineering*, pages 62–67. IEEE.

Jarke, M. and Quix, C. (2022). Federated data integration in data spaces. In *Designing Data Spaces*, pages 181–194. Springer.

Jeffery, S. R., Franklin, M. J., and Halevy, A. Y. (2008). Pay-as-you-go user feedback for dataspace systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 847–860.

Jovanovic, P., Nadal, S., Romero, O., Abelló, A., and Bilalli, B. (2021). Quarry: a user-centered big data integration platform. *Information Systems Frontiers*, 23(1):9–33.

Jovanovic, P., Romero, O., and Abelló, A. (2016). A unified view of data-intensive flows in business intelligence systems: a survey. In *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXIX*, pages 66–107. Springer.

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., and Peterson, A. S. (1990). Feature-oriented domain analysis (FODA) feasibility study. Technical report, DTIC Document.

Kantorovitch, J., Niskanen, I., Kalaoja, J., and Staykova, T. (2017). Designing situation awareness.

Khalajzadeh, H., Abdelrazek, M., Grundy, J., Hosking, J., and He, Q. (2018). A survey of current end-user data analytics tool support. In *2018 IEEE International Congress on Big Data (BigData Congress)*, pages 41–48. IEEE.

Kordjamshidi, P., Singh, S., Khashabi, D., Christodoulopoulos, C., Summons, M., Sinha, S., and Roth, D. (2017). Relational learning and feature extraction by querying over heterogeneous information networks. *arXiv preprint arXiv:1707.07794*.

Koupil, P., Hricko, S., and Holubová, I. (2022). A universal approach for multi-model schema inference. *Journal of Big Data*, 9(1):1–46.

Latih, R., Patel, A. M., Zin, A. M., Yiqi, T., and Muhammad, S. H. (2011). Whip: A framework for mashup development with block-based development approach. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pages 1–6. IEEE.

Lee, Y.-J. (2014). Semantic-based data mashups using hierarchical clustering and pattern analysis methods. *J. Inf. Sci. Eng.*, 30(5):1601–1618.

Lee, Y.-J. and Kim, J.-S. (2012). Automatic web api composition for semantic data mashups. In *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, pages 953–957. IEEE.

Li, G. (2017). Human-in-the-loop data integration. *Proceedings of the VLDB Endowment*, 10(12):2006–2017.

Liu, C., Wang, J., Han, Y., *et al*. (2015). Discovery of service hyperlinks with user feedbacks for situational data mashup. *International Journal of Database Theory and Application*, 8(4):71–80.

Löser, A., Hueske, F., and Markl, V. (2008). Situational business intelligence. In *International Workshop on Business Intelligence for the Real-Time Enterprise*, pages 1–11. Springer.

Maskat, R. (2016). *Pay-As-You-Go Instance-Level Integration*. PhD thesis, The University of Manchester (United Kingdom).

Masmoudi, M., Lamine, S. B. A. B., Zghal, H. B., Archimede, B., and Karray, M. H. (2021). Knowledge hypergraph-based approach for data integration and querying: Application to earth observation. *Future Generation Computer Systems*, 115:720–740.

Matskanis, N., Andronikou, V., Massonet, P., Mourtzoukos, K., and Roumier, J. (2012). A linked data approach for querying heterogeneous sources. pages 411–414.

Miller, R. J. (2018). Open data integration. *Proceedings of the VLDB Endowment*, 11(12):2130–2139.

Missier, P., Fernandes, A. A., Lengu, R., Guerrini, G., and Mesiti, M. (2009). Data quality support to on-the-fly data integration using adaptive query processing. In *SEBD*, pages 213–220.

Mountantonakis, M. and Tzitzikas, Y. (2019). Large-scale semantic integration of linked data: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40.

Nadal, S., Romero, O., Abelló, A., Vassiliadis, P., and Vansummeren, S. (2019). An integration-oriented ontology to govern evolution in big data ecosystems. *Information systems*, 79:3–19.

Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., and Arocena, P. C. (2019). Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment*, 12(12):1986–1989.

Nicklas, D., Schwarz, T., and Mitschang, B. (2017). A schema-based approach to enable data integration on the fly. *International Journal of Cooperative Information Systems*, 26(01):1650010.

Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., and Belfkih, S. (2018). Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4):431–448.

Paredes-Valverde, M. A., Alor-Hernández, G., Rodríguez-González, A., Valencia-García, R., and Jiménez-Domingo, E. (2015). A systematic review of tools, languages, and methodologies for mashup development. *Software: Practice and Experience*, 45(3):365–397.

Paton, N. W., Belhajjame, K., Embury, S. M., Fernandes, A. A., and Maskat, R. (2016). Pay-as-you-go data integration: Experiences and recurring themes. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 81–92. Springer.

Paton, N. W., Christodoulou, K., Fernandes, A. A., Parsia, B., and Hedeler, C. (2012). Pay-as-you-go data integration for linked data: opportunities, challenges and architectures. In *Proceedings of the 4th International Workshop*

on Semantic Web Information Management, pages 1–8.

Qi, S. and Luo, Y. (2016). Object retrieval with image graph traversal-based re-ranking. *Signal Processing: Image Communication*, 41:101–114.

Schobbens, P.-Y., Heymans, P., and Trigaux, J.-C. (2006). Feature diagrams: A survey and a formal semantics. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 139–148. IEEE.

Sehar, U., Ghazal, I., Mansoor, H., and Saba, S. (2022). A comprehensive literature review on approaches, techniques & challenges of mashup development. *International Journal of Scientific & Engineering Research*, 13.

Serban, F., Vanschoren, J., Kietz, J.-U., and Bernstein, A. (2013). A survey of intelligent assistants for data analysis. *ACM Computing Surveys (CSUR)*, 45(3):1–35.

Serrano, F. R., Fernandes, A. A., and Christodoulou, K. (2018). An approach to quantify integration quality using feedback on mapping results. *International Journal of Web Information Systems*.

Singh, V., Chen, S.-S., Singhania, M., Nanavati, B., Gupta, A., *et al*. (2022). How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–a review and research agenda. *International Journal of Information Management Data Insights*, 2(2):100094.

Tatemura, J., Chen, S., Liao, F., Po, O., Candan, K. S., and Agrawal, D. (2008). Uqbe: uncertain query by example for web service mashup. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1275–1280.

Tran, T. N., Truong, D. K., Hoang, H. H., and Le, T. M. (2014). Linked data mashups: A review on technologies, applications and challenges. In *Asian Conference on Intelligent Information and Database Systems*, pages 253–262. Springer.

Umbrich, J., Hogan, A., Polleres, A., and Decker, S. (2015). Link traversal querying for a diverse web of data. *Semantic Web*, 6(6):585–624.

Vo, Q. D., Thomas, J., Cho, S., De, P., and Choi, B. J. (2018). Next generation business intelligence and analytics. In *Proceedings of the 2nd International Conference on Business and Information Management*, pages 163–168.

Wang, G., Fang, J., and Han, Y. (2013). Interactive recommendation of composition operators for situational data integration. In *2013 International Conference on Cloud and Service Computing*, pages 120–127. IEEE.

Xu, P., Lu, J., *et al*. (2019). Towards a unified framework for string similarity joins. *Proceedings of the VLDB Endowment*.

Zhang, Y. and Ives, Z. G. (2020). Finding related tables in data lakes for interactive data science. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1951–1966.

Ziegler, P. and Dittrich, K. R. (2007). Data integration—problems, approaches, and perspectives. In *Conceptual modelling in information systems engineering*, pages 39–58. Springer.