Utility-Driven Data Management for Data-Intensive Applications in Fog Environments

Cinzia Cappiello, Barbara Pernici, Pierluigi Plebani, and Monica Vitali(✉)

DEIB, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy
{cinzia.cappiello,barbara.pernici,pierluigi.plebani,
monica.vitali}@polimi.it

Abstract. The usage of sensors, smart devices, and wearables is becoming more and more common, and the amount of data they are able to generate can create a real value only if such data are properly analyzed. To this aim, the design of data-intensive applications needs to find a balance between the value of the output of the data analysis – that depends on the quality and quantity of available data – and the performance.

The goal of this paper is to propose a "data utility" model to evaluate the importance of data with respect to their usage in a data-intensive application running in a Fog environment. This implies that the data, as well as the data processing, could reside both on Cloud resources and on devices at the edge of the network. On this basis, the proposed data utility model puts the basis to decide if and how data and computation movements from the edge to the Cloud – and vice versa – can be enacted to improve the efficiency and the effectiveness of applications.

Keywords: Data utility · Cloud and Fog computing

1 Introduction

With an increasing trend, data-intensive applications are becoming fundamental for the analysis of data gathered by the Internet of Things (IoT) [17]. In fact, data collected through tiny and affordable sensors, and transmitted with smart devices, are enabling the fourth industrial revolution supporting, for instance, predictive maintenance of machineries, real-time tracking of production lines, as well as efficient scheduling of tasks. At the same time, mobile phones and wearables are changing the habits of people, as the data collected by these devices can be exploited to optimize the daily activities and improve quality of life.

As the available amount of data increases, data-intensive applications require more and more resources to properly manage and process such data. This is witnessed by the numerous tools available to support both data transmission and data processing, where scalability is the key feature (e.g., Apache Kafka, Apache Spark, and Apache Flume). Regardless of the specific solution, efficiency in data processing is ensured by specific file systems (e.g., HDFS) enabling a proper data

S. de Cesare and U. Frank (Eds.): ER 2017 Workshops, LNCS 10651, pp. 216–226, 2017.
https://doi.org/10.1007/978-3-319-70625-2_20

management: data are spread among different nodes to enable parallel computation, replication is allowed for improving the reliability, and data formats adopt grammars enabling efficient parsing. Furthermore, the computation usually relies on resources available on the Cloud, implying the possibility to easily scale in/out the application with respect to the amount of data to be processed.

The goal of this paper is to propose a *data utility model* evaluating the usefulness of data for an application with respect to the content of the data source, and also to the quality of the content and the data source location (on the Cloud or on the Edge), which could affect the quality of data provisioning. In this paper we present the first step towards this direction: the definition of a model including all the components affecting the data utility when application is running on a Fog environment.

The rest of the paper is organized as follows. To better clarify our approach, we use the running example introduced in Sect. 2. Section 3 discusses the conceptual model of data-intensive applications running in a Fog environment. Section 4 introduces the data utility model for the previously defined data-intensive applications. Finally, Sect. 5 gives an overview of the state of the art in this field, whereas Sect. 6 concludes the paper outlining possible future work.

2 Running Example

Figure 1 draws a possible scenario, in the ambient intelligence domain, that is used along this paper. The data-intensive application to be developed analyzes the comfort in a building using the data coming from sensors placed in different rooms (e.g., temperature, humidity, and brightness sensors). The application also uses weather data, made available by external entities, to perform a validation of the sensors and to predict possible variations suggesting actions to the users through a dashboard. The tasks composing the application are: (i) *Ambient Sensing Alignment (Task A)*: it collects data coming from a set of sensors placed in the monitored building and performs some pre-processing operations, such as timestamps alignment and data cleaning. (ii) *Ambient Sensing Aggregation (Task B)*: it uses the output of the previous step to perform statistical analysis and aggregations (minimum, maximum, and average values for each sensor or for several sensors of the same kind in the building) which constitute the data set relevant for the analysis. (iii) *Data Enrichment and Prediction (Task C)*: it integrates data produced by the previous task with information about weather in the city where the monitored building is placed. (iv) *Visualization Preparation (Task D)*: it prepares the information obtained from the analysis for providing visualization tools to the final user.

Each task has different requirements in terms of data sources to be accessed. Input of a task can be the output of another task ($E_{i,j}$ representing the exchanged information from task t_i to task t_j) or a data sources that could be either directly managed by the application developer or provided by external entities. Orthogonally, data sources can be placed on devices at the edge of the network or they reside in the Cloud. In our example, *Task A* will access to data generated by a well-defined set of IoT devices (i.e., placed at the edge of the

network), producing streaming and real time information about the monitored building (Building Sensors DS_B in Fig. 1). *Task C* uses weather data sets placed in the Cloud (Weather Data DS_W in Fig. 1). While for *Task A* a specific data source is defined through the identification of the building to be monitored, for *Task C* we assume that several public data sources fit the requirements of the application. Deciding which is the best data source to be used and where to place the tasks or to move the data (i.e., at the edge or in the cloud) impacts the efficiency and the effectiveness of the application. For instance, executing the *Task A* and *Task B* on a device close to the sensors, rather than on the Cloud, could reduce the amount of data transmitted over the network and thus improve performance. At the same time, if the device on which we aim to execute those tasks has a low performance, it may become a bottleneck for the application. Moreover, the different weather data sources could have different quality levels, so the selection could finally affect the quality of the analysis in *Task C*.

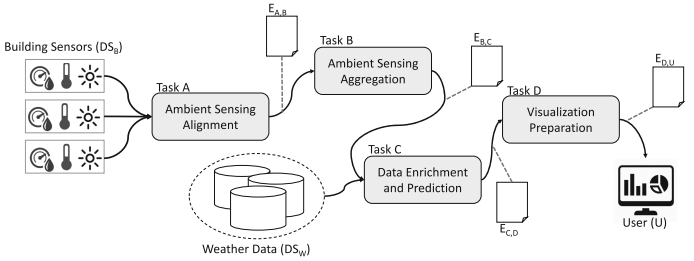

Fig. 1. Example of Data-intensive application in ambient intelligence domain

To make the designer mainly focused only on the application logic, in our approach we introduce the Data Utility as a complex metric which reflects to which extent a data source satisfies the requirements of an application. With the definition of the Data Utility we want to reduce the burden, from the developer standpoint, of selecting the data sources and the location of the tasks.

3 Data-Intensive Applications in Fog Environment

Data-intensive applications are mainly defined by the data to be processed and the computation performed on them. To this aim, a data-intensive application is often modeled with a data flow that defines the data sources that feed the application, as well as the steps to be performed to acquire, manage, and transform such data. The literature already proposes some approaches for modeling data-intensive applications. For instance, in [7] a UML profile has been specifically designed to capture the dependencies between the tasks operating on the data and the data themselves. Yet, the meta-models in [14] cover a broader spectrum: from the business view to a more technical view.

A peculiar aspect of our approach relies on the adoption of Fog computing paradigm for designing and running the application. According to the definition in [15], Fog computing builds upon the capabilities of Cloud computing, extending them toward the edge of the network. As a consequence, data-intensive applications can consider Cloud and Edge as a continuum where both data and tasks can be moved from the edge to the Cloud and vice versa. The designer, instead of specifying a precise deployment plan, specifies the characteristics that a node should, or must, have to run a task or to store data. At execution time, the deployment is adapted, while the application is running, to improve the efficiency and the effectiveness of the application. Focusing on the design standpoint, a *Data Intensive Application* designed to run on a Fog environment relies on three main elements: resources, data sources, and tasks (Fig. 2).

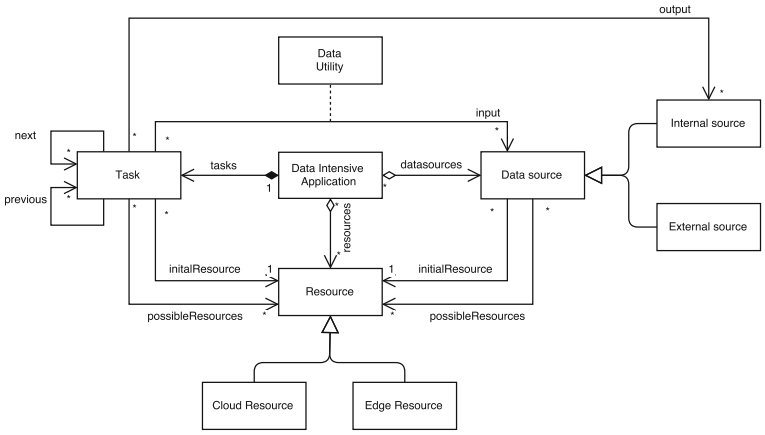

Fig. 2. Data-intensive application model.

Resource. Adopting the Fog Computing paradigm, the infrastructural layer consists of nodes (e.g., laptops, smart devices, sensors, VMs), living either on the *Cloud* or on the *Edge*, representing possible resources for our application. Regardless of the actual physical location of a resource, we are interested on its computational and storage capabilities. For an exhaustive description of the hardware characteristics of a node, standard approaches like the DMTF-CIM[1] could be adopted. Regardless of the specific model adopted, this information is required for understanding if a resource is suitable for hosting the execution of a tasks or for storing some data produced or consumed during the execution. The distinction between Cloud and Edge resources influences the level of detail with which a designer can describe a resource. A Cloud resource is managed by an external entity, i.e., the cloud provider, thus only a limited set of information

[1] http://www.dmtf.org/standards/cim.

is available for the resource. Conversely, for resources living on the Edge, we assume that the designer knows all the details about the resources and it has the ability to reconfigure or to adapt them in case modifications are required during the execution. Referring to our running example, we can assume that the designer has the possibility to change – both decreasing or increasing – the sampling rate for the sensors to a value that is optimal for the application. Conversely, the way in which the weather data are collected cannot be modified by the designer and we can also assume it is not feasible also for the Cloud provider hosting such data sources as they could be shared with other customers.

Data Source. The data source models the information needed to allow the application to read or write the relevant data. A data source is described in terms of the data content (i.e., how the data are structured), data access (i.e., how to reach data), and data utility (i.e., the relevance for the usage context). A data source is also defined by the *initialResource* where the source is initially deployed, as well as the *possibleResources* in which it could be moved as they satisfy the functional constraints required by the data source (e.g., the size, the DBMS or the file system). Inspired by the approach proposed in [7], data sources are distinguished between *Internal Source*s and *External Source*s. In the former case, we have data that are directly managed by the application designer. This category includes data produced by a task and consumed either by another task or by the final user (i.e., the $E_{i,j}$ data in the running example), data coming from resources managed by the designer (i.e., temperature sensors at the edge), or data sources on the Cloud where their management is under the responsibility of the designer (but the management of the resources on which the data source is deployed is up to the Cloud provider). In case of *External source*s, data exist independently of the application, but need to be accessed by it (e.g., the Weather data DS_W). In the proposed model, we assume that data content and data access are independent from the resource in which the data source is placed. This could be possible if a proper abstraction level is adopted to describe the intensional schema of the data, as also suggested in [2]. About the data access, using a proper naming scheme, like the one based on URI, makes the location transparency possible. Thus, when data source moves from a resource to another among the possible ones, this does not affect the behavior of tasks using these data. Data access also includes the definition of how to interact with the data source, distinguishing between stream or conventional methods. The focus of this paper concerns the *Data Utility* that measures to which extent a data source is relevant for a given usage. In the model, *Data Utility* (DU) is seen as a characterization of the association between a *Task* and a *Data Source* related to the input of the task. As discussed in Sect. 4, data utility of a data source extends the more traditional data quality concept. Differently from data content and access, DU could be affected by the resources used to host the data source. The DU model introduced in the next section makes this dependency explicit.

Task. A task represents a unit of work to be performed during the execution of the application. As suggested in [14], especially in data-intensive applications,

there is a set of common tasks like data cleaning, data integration, data compression, data encryption, and the application can be seen as a composition of such tasks. In some cases, the algorithms behind those tasks are well known (e.g., clustering, regression), in some others the designer has to produce some scripts implementing the custom data processing. As for the data sources, there is a set of *possibleResources* on which the task can be deployed and one of them represents the *initialResource* where the task is initially deployed.

We assume that tasks are organized according to a data-flow process [5] which highlights how data are acquired, transformed, and returned. This flow is defined using *next* and *previous* attributes which capture the possible execution flows. The connection between the tasks and the data sources represents the input or the output of the tasks. In case of *output*, the task is connected to the storage nodes, as they represent data produced internally to the application. On the other side, the *input* of a task can be modeled both as a storage node or a source node as the input of a task can be data produced by a preceding task or made available by an external element.

4 Data Utility Model

Based on the data-intensive application model introduced in Sect. 3, now we go into the details of some of the concepts which constitute the elements for defining the Data Utility.

Starting from the available data sources, we indicate them with:

$$DS = \{ds_j\} = \{< S_j, ir_j, PR_j >\}$$

where S_j is the data source schema, ir_j is the resource on which ds_j is initially deployed, and PR_j is the set of possible resources on which it could be deployed. As the initial resource is, by definition, one the possible resource then $ir_j \in PR_j$.

Moving to the tasks that compose the data-intensive application, we assume that each task t_i is defined by:

$$t_i =< D_i, IN_i, OUT_i, P_i, N_i, ir_i, PR_i >$$

where *(i)* D_i is a description of the task in terms of type of operations performed (e.g., aggregation, filtering, clustering, association), *(ii)* IN_i and OUT_i are the sets of task inputs and outputs, *(iii)* P_i and N_i are the set of tasks that precede and follow the analyzed task in the data-flow process, *(iv)* ir_i refers to the initial resource on which the task is deployed, while PR_i the set of resources on which it can be potentially deployed. Similarly to what stated for the data sources, $ir_i \in PR_i$.

As described in Sect. 2, tasks may gather inputs (i) from a specific data source (i.e., *Task A*), (ii) from a previous task (i.e., *Task B*) and (iii) from a data source that should be selected from a set of candidate sources (i.e., *Task C*). Note that we consider cases (i) and (ii) as equivalent since we assume that the output of a task can be seen a data source. Furthermore, case (iii) is the situation in which the developer should be supported in the selection of the sources.

A task t_i may have several inputs. The k-th input is defined as:

$$IN_{ik} =< A_{ik}, CDS_{ik} >$$

where, A_{ik} is the set of the attributes of the data source required by the task (e.g., temperature, humidity), and $CDS_{ik} \subseteq DS$ the set of candidate data sources from which data have to be extracted, which can be both internal and external sources. If $|CDS_{ik}| = 1$, it means that the designer has specified the specific source to consider; otherwise the designer would like to be supported in the identification of the most suitable source in the specified set. In this last situation, the developer specifies a request R_{ik} over the input IN_{ik} in order to provide all the elements that can affect the source selection. Let us define the request as:

$$R_{ik} =< IN_{ik}, f^*_{ik}, NF^*_{ik} >$$

where, f_{ik} is an optional parameter to express functional requirements, while NF_{ik} is an optional parameter expressing a set of required non-functional properties. More precisely, f_{ik} is a predicate composed of atoms linked by traditional logical operators (i.e., AND, OR) that allows developers to specify restrictions over the allowed values in order to better drive the source selection (e.g., city= "Milan" AND Temp > 23). On the other hand, NF_{ik} contains requests related to DQ (Data Quality) or QoS (Quality of Service) aspects. The former focuses on the quality of the content provided by the source, while the latter regards performance issues such as availability and latency.

It is worth noting that the satisfaction of functional requirements only depends on the content of the data source, whereas the satisfaction of the QoS constraints depends on the resources on which the task or the data are deployed/stored. Therefore the suitability of a data source has to be specified by considering not only the data it contains but also the execution environment.

In this paper, for defining this suitability we introduce the Data Utility concept. *Data Utility (DU)* can be defined as *the relevance of data for the usage context*, where the context is defined in terms of the designer's goals and system characteristics. The designer's goals are captured by the definition of t_i which includes the input descriptions and the related requests in terms of both functional and non-functional requirements, while the system characteristics include the definition of the data sources DS. On these basis, Data Utility of a data source ds_j for a task t_i is defined as:

$$DU_{ixjy} = f(< t_i, r_x >, < ds_j, r_y >)$$

Since both tasks and data sources can be placed on different resources belonging to PR_i and PR_j, respectively, data utility depends on the task ($< t_i, r_x >$) and data sources ($< ds_j, r_y >$) placement, where $r_x \in PR_i$ and $r_y \in PR_j$.

We assume that the data sources are associated with a set of metadata that reveal the *Potential Data Utility* (PDU) that summarizes the capabilities of the data source and can be periodically evaluated independently of the context. The PDU is calculated looking at the data and the characteristics of the data

source. It is derived from a *Data Quality* and a *Reputation* assessment. Generally speaking, as in [19] data quality can be defined as the fit for use for a data consumer and it implies a multi-dimensional analysis including dimensions like accuracy, completeness, timeliness [1]. In fact, errors, missing, or updated data affect the usage and potential benefits of data. The assessment of Data Quality dimensions may contribute to the understanding of the potential value of the data. The list of dimensions and the assessment metrics depend on the type of data contained in the source. For example, sensors data need the evaluation of additional attributes such as precision and data stability and algorithms for evaluating accuracy change along the type of data (i.e., strings vs. numeric values). Anyway, we can assume that each source is associated with a set of Data Quality dimensions and related values. Besides the content, also the history about the usage of the source should be considered. For this reason we define a Reputation index as the likelihood that a data source will satisfy the application requirements. For now, we compute the reputation by considering the frequency with which the source has been used and the respective success rate, and the scope of data (e.g., generic or specific, integrable with other sources, used with other sources). PDU provides an objective way to rank similar sources and can be useful for a pre-filtering of the sources. However, a data source has to be evaluated by considering the context that in our scenario is composed of the data-intensive application and the available resources. Given a request R_{ik} together with the characteristics of the task and the set of candidate data sources, the request can be enriched with additional data quality constraints derived by the type of task (e.g., data mining operations requires a high amount of data and completeness). The task type and request may also force the recalculation of the some data quality dimensions (i.e., if the request is limited to a subset of attributes of the source, the quality should be evaluated only on the considered data set).

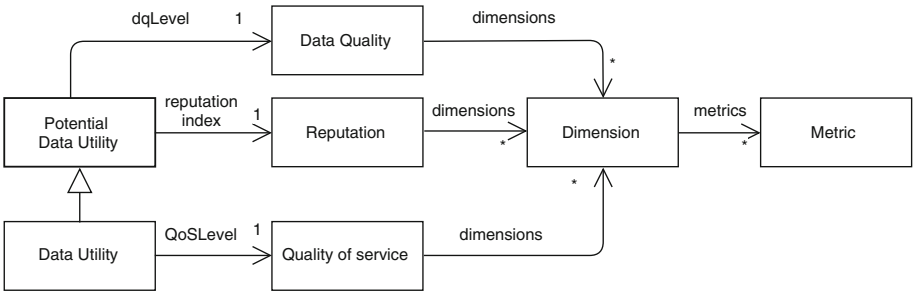

Fig. 3. Model of the utility components

QoS capabilities have to be evaluated by considering all the available options that the Fog environment offers. Thus, both tasks and data can be moved from edge to cloud and vice-versa, from edge to edge or through cloud resources and that the placement of a task or a data source on a specific resource has surely

an impact on the QoS: in fact, the computational cost for obtaining data and the latency changes on the basis of the chosen location. Therefore, we calculate the QoS dimensions for each possible configuration defined in terms of task placement $< t_i, r_x >$ and data placement $< ds_j, r_y >$.

In summary, Data Utility can be assessed by considering three main aspects (Fig. 3): Data Quality, Reputation, and Quality of Service. Each of them is evaluated by means of dimensions, each one associated with different metrics (more than one assessment function might be available for a single dimension). Discarding the sources and configurations that do not satisfy functional and non-functional requirements, it is possible to associate with each source ds_j belonging to CDS_{ik} different Data Utility indicators (one for any admissible configuration), each one expressed as a set of three indices: Data Quality, Reputation and QoS.

5 Related Work

Managing data, meta-data and their storage and transformation has been addressed in several areas of research focusing on a number of separate though possibly interrelated aspects. Data utility has been defined in different ways in the literature. In statistics data utility is defined as "A summary term describing the value of a given data release as an analytical resource. This comprises the data's analytical completeness and its analytical validity" [9]. In business scenarios data utility is conceived as "business value attributed to data within specific usage contexts" [16] while in IT environments it has been described as "The relevance of a piece of information to the context it refers to and how much it differs from other similar pieces of information and contributes to reduce uncertainty" [11]. All these definitions agree on the dependency of DU on the context in which data are used. Therefore, the assessment of DU is a complex issues since context can be composed of several elements and it usually changes over time. Early studies in data utility assessment have been carried out in the area of information economics, investigating information utility mainly from a mathematical perspective. In [20], the relevant economic factors for assessing DU are: (i) the costs and benefits associated with obtaining data, (ii) the costs associated with building the analysis algorithm to process data, and (iii) the costs and benefits derived from utilizing the acquired knowledge. Later, the growing adoption of IT in business shifted the attention towards the information utility associated with business processes [4]. Other papers analyze DU by considering some specific usage context such as data mining applications [12] or by considering context as limited to users requirements [10]. Another important contribution relates DU to information quality dimensions, e.g. accuracy and completeness [13]. Information quality requirements for obtaining valuable results from processes have been discussed in [6]. In [18], data utility is discussed in mobile clouds with the focus on optimizing the energy efficiency of mobile devices. Energy efficiency was also the focus of [8] where the interrelations between data value evaluation and adaptation strategies have been discussed, with a focus on run-time adaptation rather than on the design of applications.

We propose a more comprehensive definition of Data Utility that includes both the content of the data sources, the application using them and the execution environment, considering all the data and computation movement actions that fog computing enables. Note that the concept of data movement has been discussed in [3] as a basis for providing operations for improving quality of data and service. However, while that paper focuses on possible operations and strategies, in the current paper we focus on a comprehensive evaluation of data utility.

6 Concluding Remarks

In this paper we have introduced a conceptual model to define the Data Utility for data-intensive applications in a Fog environment. The proposed model takes into account the relationship between the tasks composing the application and the data sources that can be used by such tasks to perform the required computation. As the location of both tasks and data sources can change, the influence of data and computation movement is considered in the Data Utility Model. The evaluation of the Data Utility and the definition of a Global Utility Model for the whole application is under investigation to consider the influences among tasks, as well as the constraints over the deployment.

Acknowledgments. DITAS project is funded by the European Union Horizon 2020 research and innovation programme under grant agreement RIA 731945.

References

1. Batini, C., Scannapieco, M.: Data and Information Quality - Dimensions, Principles and Techniques. Data-Centric Systems and Applications. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-24106-7
2. Cleve, A., Brogneaux, A.-F., Hainaut, J.-L.: A conceptual approach to database applications evolution. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 132–145. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16373-9_10
3. D'Andria, F., Field, D., Kopaneli, A., Kousiouris, G., Garcia-Perez, D., Pernici, B., Plebani, P.: Data movement in the Internet of Things domain. In: Dustdar, S., Leymann, F., Villari, M. (eds.) ESOCC 2015. LNCS, vol. 9306, pp. 243–252. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24072-5_17
4. Even, A., Shankaranarayanan, G., Berger, P.D.: Inequality in the utility of customer data: implications for data management and usage. J. Database Mark. Custom. Strat. Manag. **17**(1), 19–35 (2010)
5. Garijo, D., Alper, P., Belhajjame, K., Corcho, Ó., Gil, Y., Goble, C.A.: Common motifs in scientific workflows: an empirical analysis. Future Generat. Comp. Syst. **36**, 338–351 (2014)
6. Gharib, M., Giorgini, P., Mylopoulos, J.: Analysis of information quality requirements in business processes, revisited. Requir. Eng. 1–23 (2016)
7. Gómez, A., Merseguer, J., Di Nitto, E., Tamburri, D.A.: Towards a UML profile for data intensive applications. In: Proceedings of the International Workshop on Quality-Aware DevOps, Saarbrücken, Germany, pp. 18–23 (2016)

8. Ho, T.T.N., Pernici, B.: A data-value-driven adaptation framework for energy efficiency for data intensive applications in clouds. In: 2015 IEEE Conference on Technologies for Sustainability (SusTech), pp. 47–52. IEEE (2015)
9. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E.S., Spicer, K., de Wolf, P.P.: Statistical Disclosure Control. Wiley, New York (2012)
10. Ives, B., Olson, M.H., Baroudi, J.J.: The measurement of user information satisfaction. Commun. ACM **26**(10), 785–793 (1983)
11. Kock, N.F.: Encyclopedia of E-collaboration. Information Science Reference - Imprint of: IGI Publishing, Hershey (2007)
12. Lin, Y.C., Wu, C.W., Tseng, V.S.: Mining high utility itemsets in big data. In: Cao, T., Lim, E.P., Zhou, Z.H., Ho, T.B., Cheung, D., Motoda, H. (eds.) PAKDD 2015. LNCS, vol. 9078, pp. 649–661. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18032-8_51
13. Moody, D., Walsh, P.: Measuring the value of information: an asset valuation approach. In: European Conference on Information Systems (1999)
14. Nalchigar, S., Yu, E., Ramani, R.: A conceptual modeling framework for business analytics. In: Comyn-Wattiau, I., Tanaka, K., Song, I.Y., Yamamoto, S., Saeki, M. (eds.) ER 2016. LNCS, vol. 9974, pp. 35–49. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_3
15. OpenFog Consortium Architecture Working Group: OpenFog Architecture Overview, February 2016. http://www.openfogconsortium.org/ra
16. Syed, M.R., Syed, S.N.: Handbook of Research on Modern Systems Analysis and Design Technologies and Applications. Information Science Reference - Imprint of: IGI Publishing, Hershey (2008)
17. Turner, V., Reinsel, D., Gatz, J.F., Minton, S.: The digital universe of opportunities. IDC White Paper, April 2014
18. Wang, J., Zhu, X., Bao, W., Liu, L.: A utility-aware approach to redundant data upload in cooperative mobile cloud. In: 9th IEEE International Conference on Cloud Computing, CLOUD 2016, San Francisco, CA, USA, pp. 384–391 (2016)
19. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. J. Manage. Inf. Syst. **12**(4), 5–33 (1996)
20. Weiss, G.M., Zadrozny, B., Saar-Tsechansky, M.: Guest editorial: special issue on utility-based data mining. Data Min. Knowl. Discov. **17**(2), 129–135 (2008)