User-Centred tooling for the Modelling of Big Data Applications

Hourieh Khalajzadeh
hourieh.khalajzadeh@monash.edu
Monash University
VIC, Australia

Tarun Verma
tver0005@student.monash.edu
Monash University
VIC, Australia

Andrew J. Simmons
a.simmons@deakin.edu.au
Deakin University
VIC, Australia

John Grundy
john.grundy@monash.edu
Monash University
VIC, Australia

Mohamed Abdelrazek
mohamed.abdelrazek@deakin.edu.au
Deakin University
VIC, Australia

John Hosking
j.hosking@auckland.ac.nz
University of Auckland
Auckland, New Zealand

ABSTRACT

We outline the key requirements for a Big Data modelling recommender tool. Our web-based tool is suitable for capturing system requirements in big data analytics applications involving diverse stakeholders. It promotes awareness of the datasets and algorithm implementations that are available to leverage in the design of the solution. We implement these ideas in BiDaML-web, a proof of concept recommender system for Big Data applications, and evaluate the tool using an empirical study with a group of 16 target end-users. Participants found the integrated recommender and technique suggestion tools helpful and highly rated the overall BiDaML web-based modelling experience. BiDaML-web is available at https://bidaml.web.app/ and the source code can be accessed at https://github.com/tarunverma23/bidaml.

CCS CONCEPTS

• Software and its engineering → Visual languages; Requirements analysis; • Information systems → Data mining; • Human-centered computing → Human computer interaction (HCI).

KEYWORDS

big data applications, BiDaML, recommender

ACM Reference Format:
Hourieh Khalajzadeh, Tarun Verma, Andrew J. Simmons, John Grundy, Mohamed Abdelrazek, and John Hosking. 2020. User-Centred tooling for the Modelling of Big Data Applications. In *MODELS '20: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, October 18–23, 2020, Montreal, QC.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

As business is increasingly conducted electronically online, companies and organisations have access to a wealth of big data. Successfully leveraging these data for decision making requires teams of domain experts, data scientists and software engineers to design and build the right big data system. Due to the high-stakes interdisciplinary nature of big data projects, a suitable requirements modelling language and associated tooling is needed to capture a shared model of the big data system amongst all stakeholders.

Modelling languages such as UML and BPMN help to formally capture software architecture and business process requirements for traditional software engineering projects. However, these modelling languages are targeted at professional software developers, and their emphasis on formalism for computers rather than cognitive effectiveness for humans has led to criticism, particularly for novice users without a background in software engineering.

As big data projects have their own set of concerns beyond traditional software engineering projects and require engagement from diverse stakeholders [7, 15], there is a need for domain specific modelling languages and associated tooling to address the needs of big data projects. One such attempt is the Big Data Analytics Modeling Languages (BiDaML) [6], which provides a metamodel and associated set of visual notations for end-to-end requirements analysis of big data problems, and was designed to be comprehensible to a diverse set of stakeholders. However, while users responded positively to the language and notations themselves, they responded poorly to the tool used to support the modelling language.

To address these shortcomings, in this paper we propose a user-centered approach to the design of tooling for modelling of big data systems. We begin with a motivating example, then outline key requirements for big data modelling systems from a user-centred design perspective. We review background in each area and highlight gaps in existing approaches. We implement a proof of concept web-based tool, and explain how the requirements were used to guide our design choices. Finally, we conduct a user study to evaluate our approach with end-users.

2 MOTIVATING EXAMPLE

Lee is a business analyst working with a team of data scientists and software engineers to build a tool to assist in epileptic seizure detection. To ensure that the tool meets clinical needs, he meets with clinicians at a hospital to elicit requirements and propose a solution that will be deployed within intensive care units. However, without a shared language, he risks his designs being misinterpreted by clinicians and rest of his team. Lee selects BiDaML for this purpose; however, he finds it tedious to create BiDaML diagrams using the official tool developed using [14], and others on his team resort to manipulating the diagrams as images rather than in source form, thus preventing the diagrams from being used as a source for code generation.

The team discuss the idea of using deep learning approaches to detect epileptic seizures, and Lee adds this to the solution diagram. However, Lee is uncertain whether this approach is feasible given the small size of the dataset and the accuracy requirements that would need to be met prior to use in a clinical setting. After an initial exploration phase, Lee learns that while the data held by the hospital is insufficient alone, there are public datasets and algorithms available that could have been utilised in the design of an alternative solution. However, as Lee was not initially aware of their existence, it did not occur to him to perform a data or code search earlier.

3 BACKGROUND

3.1 Big Data Analytics

Big data analytics, as a very active area in research and industry [8, 11] aims to improve decision-making through collecting, analysing, and processing big data. It brings together stakeholders with a variety of skill-sets, technical and non-technical knowledge and backgrounds to communicate and collaborate in multidisciplinary teams. Data analytics projects involve many steps such as classifying the problem, acquiring data, processing data, modeling the problem, validation and execution, deploying, etc [12]. In fact, data processing and Machine Learning (ML) tasks, that are commonly considered as the main part of data analytics applications, are only a very small component in the building blocks necessary to build real-world deployable data analytics software systems [13].

3.2 Data Analytics Tools

There are many data analytics tools now available, such as Azure ML Studio, Amazon AWS ML, Google Cloud ML, and BigMl as reviewed in [3, 5]. Many current big data analytics tools provide only low-level data science solution design, despite many other steps being involved in solution development. These tools only cover a limited set of phases of data related operations (DataOps), AI and machine learning related operations (AIOps), and deployment and development related operations (DevOps) and none cover business problem description, requirements analysis and design (BusinessOps). Moreover, most end-users in multidisciplinary teams have limited technical knowledge of data science and programming and thus usually struggle to use these tools. Therefore, a high-level presentation of the steps to capture, represent and communicate the business requirements analysis and design, data pre-processing, high-level data analysis process, solution deployment, and data visualisation is essential [6].

3.3 Recommender Systems

Open data portals and Data search tools such as Google Datasets[1] and Kaggles Datasets [2] curate a collection of datasets that users can search through. Tools such as Papers with Code [3] attempt to provide a convenient means for users to find related papers and the code for algorithms.

[1]https://datasetsearch.research.google.com/
[2]https://www.kaggle.com/datasets
[3]https://paperswithcode.com/

Recommender tools attempt to recommend items of interest to a user based on their behaviour. Users are not always able to formulate their search explicitly, either because the idea is not fully formed, or because they lack the appropriate vocabulary to express it using terms that are indexed. In contrast, recommender systems can surface results based on data passively collected from the user.

Recommender algorithms are ranked on dimensions such as correctness, trustworthiness, robustness, usability, novelty, learning rate, user preference, diversity, serendipity, etc [1]. However, when integrated as part of a system, an important aspect is serendipity — the ability to surface items that are both useful and unexpected to a user at an appropriate time, which they may not have otherwise considered. Unfortunately, while serendipity is a much sought-after property, its subjective nature makes it difficult to measure. Nevertheless, certain strategies can help maximise serendipity [9].

4 REQUIREMENTS FOR BIG DATA MODELLING SYSTEMS

We have identified four key requirements critical for Big Data modelling, spanning the modelling, information systems, and Human-Computing Interface literature.

Model-Driven: Informal diagrams such as mind maps, flowcharts and textual descriptions can be helpful communication tools in the design phase. However, they risk ambiguities and inconsistencies that are only discovered later in the project. They then often need to be thrown away and implemented from scratch as code in the development phase. Model-driven development approaches can assist in formalising the system design requirements from the very start of the project, verifying their consistency, and tracing development decisions back to the initial requirements. Furthermore, it can eliminate some of the development altogether via code generation, thus speeding up development and allowing the team to focus on modelling of requirements and domain knowledge rather than software implementation.

Cognitively Effective: Many existing visual modelling languages, such as UML and BPMN, focus on the semantics of the language, but leave the visual notation as an afterthought, with little-to-no justification of the design choices. However, as Moody argues in [10], *notation matters*—the choice of notations impacts the ability of novice users to effectively reason about models expressed in the notation without experiencing cognitive overloaded due to the complexity of interpreting the notation itself. This is particularly important in the design of big data applications involving multidisciplinary teams in which it is important that all stakeholders are able to participate in the design of the system.

User-Centered: The use of Model-Driven Development requires tool support to formally specify the model. However, this leads to complex modelling tools with interfaces designed around the modelling approach rather than around the user. This adds friction to the design process that may explain the tendency for practitioners to resort to informal diagramming tools despite the theoretical benefits offered by formal modelling tools. In contrast, User-centred design considers the needs of the user first, then designs the system around the desired interactions with the user.

Recommendation and Decision Support: Designers are faced with the challenge of identifying and choosing between an ever growing set of datasets, algorithms and tools to support design of their big data application. As datasets, algorithms and tools are the very essence of a big data system, it is important that they are considered from the onset of the project, and that stakeholders are aware of these relationships and how they impact on the feasibility of the system. Indeed, it is often the presence of a particular dataset, or the rise of a particular class of algorithms (such as deep learning) and associated tools (such as Tensorflow), that motivate a company to consider embarking on a big data project. However, if the datasets and algorithms are not well aligned (e.g. a company executive who wants to take advantage of recent advances in deep learning to drive their decision making, but fails to understand the volume of data required for this approach to be successful), then the project will inevitably fail. As such, modelling systems for big data applications need to incorporate recommendation tools to promote an awareness of data, algorithms and tools relevant to solving the stakeholders' problem, and to help stakeholders make informed choices prior to embarking on a costly big data project.

5 OUR APPROACH

We outline development of a web-based prototype based on the four key requirements outlined in section 4. While the specific technology choices made in the implementation of our prototype are not the only way of satisfying these requirements, our intent is to highlight how they guided our decisions and to provide a baseline implementation for future research to improve upon.

5.1 BiDaML

Big Data Analytics Modeling Languages (BiDaML) is a a set of domain-specific visual languages using five diagram types at different levels of abstraction to support key aspects of big data analytics [4, 6]. These five diagram types cover the whole of the data analytics software development life cycle from higher-level requirement analysis and problem definition through the low-level deployment of the final product. List of notations for different diagram types are shown in Figure 1. The five diagrammatic types are:

- **Brainstorming diagram** provides an overview of a data analytics project and all the tasks and sub-tasks involved in designing the solution at a very high level;
- **Process diagram** specifies the analytics processes/steps including key details related to the participants (individuals and organizations), operations, and conditions in a data analytics project;
- **Technique diagrams** show the step by step procedures and techniques used for each task in the brainstorming and process diagrams at a low level of abstraction;
- **Data diagrams** document the data, artifacts, and outputs produced in each of the above diagrams at a low level;
- **Deployment diagram** depicts the run-time configuration for development related tasks.

BiDaML supports aspects of *Model-Driven* software development, such as validation rules for how elements may be connected, automatic report generation, and (limited) functionality for code generation from a diagram. In contrast to other modelling languages such as UML and BPMN, BiDaML was designed from the ground up

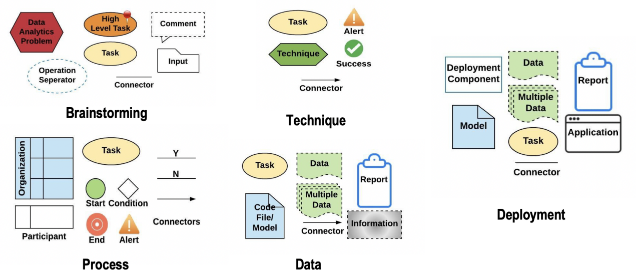


Figure 1: BiDaML 2.0 Notations [6]

using the Physics of Notations [10] framework to be *Cognitively Effective* for novice users, and has been validated though user studies [6] to be suitable for a diverse audience.

5.2 BiDaML Tooling

While users have reacted favourably to the BiDaML concept and notations themselves [4], the existing MetaEdit+ based implementation of BiDaML suffered from several serious limitations that caused users to respond negatively to the tool itself. Our MetaEdit+ based implementation was not user-centred—the interface though which users create BiDaML models was embedded within the general-purpose MetaEdit+ tool used to create the notation itself, with limited concern for users' ability to understand the interface yet alone use it efficiently—nor did it include any form of recommendation to assist users in the design of the system.

In contrast, our web-based implementation of BiDaML[4] has been designed to place the user's needs first. The diagram editor is based upon the auto-layout web-based tool vue-graphViz[5][2] which includes features to adjust placement of items to avoid clutter (such as lines crossing symbols on the graph) in order to improve readability. We include a set of quick-start questions to help the user rapidly generate the initial diagram with minimal clicks, then provide a minimal interface through which the user can modify the diagram as needed. To increase the user's awareness of relevant algorithms and datasets specific to their problem, we utilise Papers with Code and Google/Kaggle Datasets Search for recommending algorithms and datasets. Finally, our tool includes a technique recommender in order to help end users decide which techniques are appropriate given their dataset, and to consider questions such as the type of prediction or classification task, and whether they have access to sufficient labelled data.

5.3 A Web-based, Auto-layout Modelling Tool

Automated layout tools help optimally route edges and nodes to avoid cluttering and overlap of edges that compromise the readability of the diagram. We base our approach on vue-graphViz. In contrast to the original approach which allows users to design using ad-hoc notations, our system provides users with the tools to express their diagrams using BiDaML, a formalised modelling language for design of Big Data systems. Hosting of the application,

[4]https://github.com/tarunverma23/bidaml
[5]https://github.com/yusufades/vue-graphViz

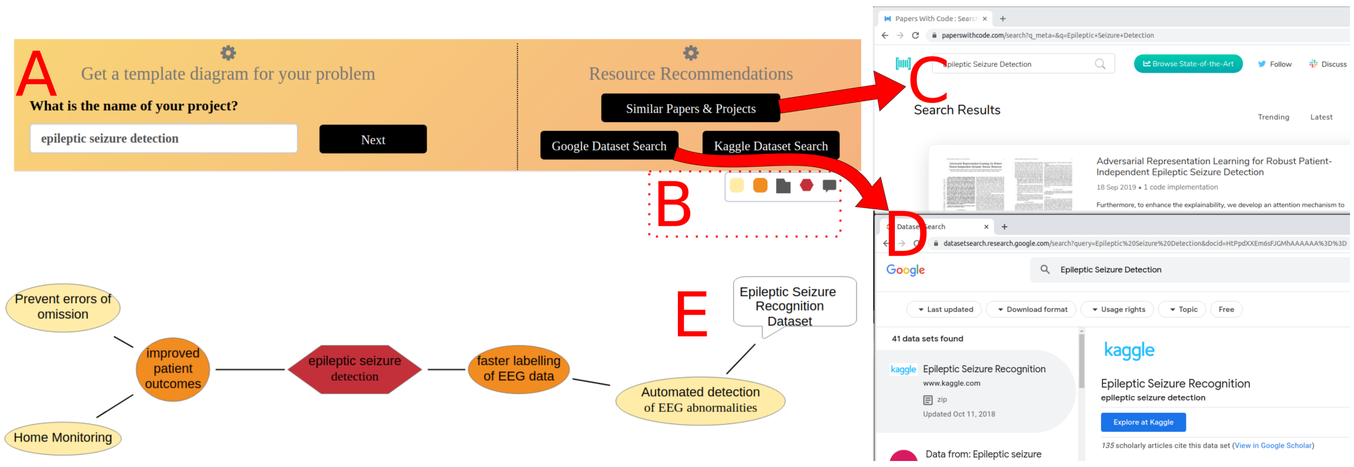

A
Get a template diagram for your problem
What is the name of your project?
epileptic seizure detection
Next
Resource Recommendations
Similar Papers & Projects
Google Dataset Search
Kaggle Dataset Search
B
C
Papers With Code : Search
paperswithcode.com/search?q_meta=&q=Epileptic+Seizure+Detection
Epileptic Seizure Detection
Browse State-of-the-Art
Follow
Discuss
Search Results
Trending
Latest
Adversarial Representation Learning for Robust Patient-Independent Epileptic Seizure Detection
18 Sep 2019 • 1 code implementation
Furthermore, to enhance the explainability, we develop an attention mechanism to
D
Dataset Search
datasetsearch.research.google.com/search?query=Epileptic%20Seizure%20Detection&docid=HtPpdXXEm6sFJGMhAAAAAA%3D%3D
Google
Epileptic Seizure Detection
Last updated
Download format
Usage rights
Topic
Free
41 data sets found
kaggle
Epileptic Seizure Recognition
www.kaggle.com
zip
Updated Oct 11, 2018
Data from: Epileptic seizure
kaggle
Epileptic Seizure Recognition
epileptic seizure detection
Explore at Kaggle
135 scholarly articles cite this data set (View in Google Scholar)
E
Epileptic Seizure Recognition Dataset
Prevent errors of omission
improved patient outcomes
epileptic seizure detection
faster labelling of EEG data
Automated detection of EEG abnormalities
Home Monitoring

Figure 2: Project template and Code/Data Recommendations

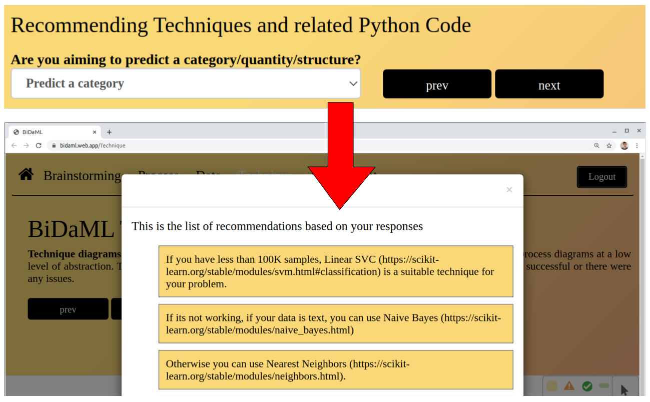
Real-time database, Google Analytics, and authentication services are implemented using Firebase's APIs. The web based interface used for designing a brainstorming diagram for our motivating example in Section 2, epileptic seizure detection, is shown in Figure 2.

5.4 Recommender Tool

Goal modelling languages such as i* help capture business or organisation goals and understand their interplay on other systems. Requirements analysis is an essential aspect of a project that determines its success. In BiDaML-web, we provide users with a brainstorming diagram where they enter their problem definition as a red hexagon in the centre of the screen. To speed up the time taken to create an initial diagram, we provide users with a diagram generation tool. The tool is equipped with two recommenders.

In contrast to the original BiDaML tool [6], we wanted BiDaML-web to provide an integrated environment for design of big data systems. BiDaML-web includes both a paper and dataset recommender tool integrated into the environment itself. This promotes users to utilise existing datasets and algorithms where available. To quick-start the process of creating a diagram, the system offers the user with prompts for the problem, objectives, and users to auto-generate an initial diagram through the "Get a template diagram for your problem" option (Figure 2A). After generation, the user can interactively add additional objects to the diagram by selecting one of the element types (task, high-level tasks, input, problem, comment) (B). The environment provides the user with easy access to algorithm implementations (C) and datasets relevant to their problem (D) that they can add to their brainstorming diagram (E). Our tool includes a technique recommender inspired by the scikit-learn algorithm cheat-sheet[6], illustrated in Figure 3. As future work, we intend to incorporate additional techniques over time. Here, based on the evolving BiDaML-Web model, a set of possible solution techniques have been found, ranked and are being suggested to the modeller. We are working on linking this to the problem definition to automatically recommend similar work as the user edits the diagram.

[6]https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html


Recommending Techniques and related Python Code

Are you aiming to predict a category/quantity/structure?

Predict a category

This is the list of recommendations based on your responses

If you have less than 100K samples, Linear SVC (https://scikit-learn.org/stable/modules/svm.html#classification) is a suitable technique for your problem.

If its not working, if your data is text, you can use Naive Bayes (https://scikit-learn.org/stable/modules/naive_bayes.html)

Otherwise you can use Nearest Neighbors (https://scikit-learn.org/stable/modules/neighbors.html).

Figure 3: Technique Recommendation

6 EVALUATION RESULTS

To evaluate BiDaML-web, we performed a user study with a group of 16 end-users. Given we had conducted comprehensive evaluations of BiDaML notations and its comparison with other modelling tools in our previous work [6], we only evaluated the auto-layout web based user interface and the recommender tools in this study. Our aim was to evaluate the usability of BiDaML-web and whether users paid attention to recommendations and found the code, paper, project and dataset recommendations, helpful.

6.1 User Study

In this study, we first introduced the BiDaML concept, notations, and diagrams and then asked a group of 16 data analysts, data scientists, domain experts and software engineers to use BiDaMl-web tool to model and describe a project of their choice. We finally asked participants to fill in a questionnaire and asked to rate whether BiDaML-web is easy to understand/learn/use and how they found the recommender tool. The group study consisted of 9 PhD students, and 7 academic staff. 8 participants categorised themselves as software engineers, 4 as data analysts/scientists, 5 as Domain expert/business analyst/business manager, and 2 as "other". The distribution of data analytics/data science experience was: 7 participants with less than 1 year; 2 participants with 2 years; 2 participant with 3 years; and 5 participants with 5 to 9 years. The distribution of programming experience was: 2 participants with 0-1 year, 1 participant with 1 year, 3 participants with 2 years, 2 participants with 3 years, 2 participants with 4 years, 2 participants with 5 to 9 years, and 4 participants with 10 or more years.

6.2 Results analysis

Study participants found the integrated recommender tools helpful, as shown in Figure 4. They also repsonded positively to the tool overall, as shown in Figure 5. The primary reasons selected were "it made me think of details that I never noticed" (9 of 16) and "introduced resources I wasn't aware of" (9 of 16); it was possible for a participant to select multiple reasons or provide a custom response to this question.

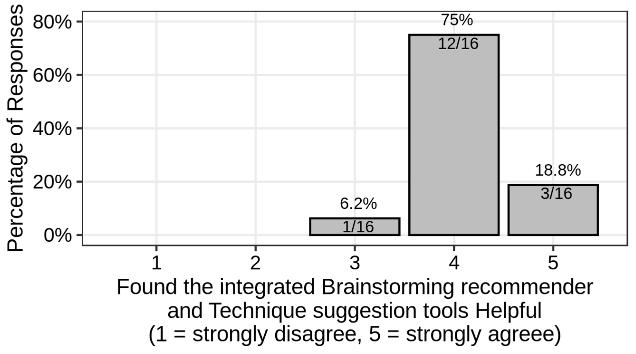


Figure 4: Participants' perception of the helpfulness of the integrated recommender and technique suggestion tools

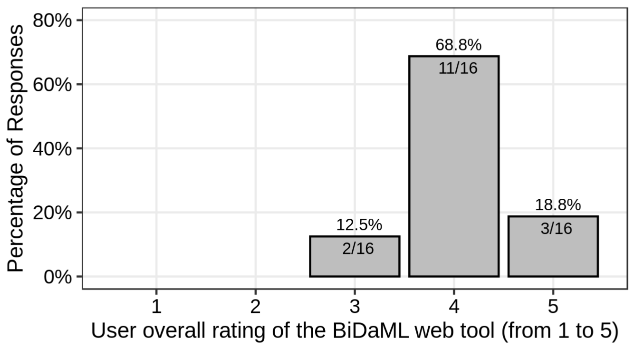


Figure 5: Participants' perception of the tool overall

6.3 Future Research

Currently our BiDaML-Web tool does not validate that the diagrams are valid BiDaML, and it does not support code generation. However, these features are present in the original BiDaML tool and thus already proved to be feasible. As future work, we could either re-implement the validation rules in JavaScript, or translate the exported models into a format that can be loaded into the original MetaEdit+ based BiDaML tool for validation and code generation.

7 CONCLUSIONS

BiDaML-web and recommender tools are presented and evaluated as a user-centric extension for our BiDaML modeling suite. Positive feedback we received from the participants encouraged us to merge the auto-layout interface and recommender tools with our modelling suite in the future. With merging them, we aim to support generating source code, reports and documentation for developing data analytics solutions.

ACKNOWLEDGMENTS

Support for part of this work from ARC Laureate Program FL190100035 is gratefully acknowledged.

REFERENCES

[1] Iman Avazpour, Teerat Pitakrat, Lars Grunske, and John Grundy. 2014. Dimensions and metrics for evaluating recommendation systems. In *Recommendation systems in software engineering*. Springer, 245–273.
[2] Tim Dwyer, Kim Marriott, and Michael Wybrow. 2008. Dunnart: A constraint-based network diagram authoring tool. In *International Symposium on Graph Drawing*. Springer, 420–431.
[3] Hourieh Khalajzadeh, Mohamed Abdelrazek, John Grundy, John Hosking, and Qiang He. 2018. A survey of current end-user data analytics tool support. In *2018 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 41–48.
[4] Hourieh Khalajzadeh, Mohamed Abdelrazek, John Grundy, John Hosking, and Qiang He. 2019. BiDaML: A Suite of Visual Languages for Supporting End-User Data Analytics. In *2019 IEEE International Congress on Big Data (BigDataCongress)*. IEEE, 93–97.
[5] Hourieh Khalajzadeh, Mohamed Abdelrazek, John Grundy, John G Hosking, and Qiang He. 2019. Survey and Analysis of Current End-user Data Analytics Tool Support. *IEEE Transactions on Big Data* (2019).
[6] Hourieh Khalajzadeh, Andrew Simmons, Mohamed Abdelrazek, John Grundy, John Hosking, and Qiang He. 2020. An End-to-End Model-based Approach to Support Big Data Analytics Development. *Journal of Computer Languages* (2020), 100964.
[7] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. 2017. Data scientists in software teams: State of the art and challenges. *IEEE Transactions on Software Engineering* 44, 11 (2017), 1024–1038.
[8] Sara Landset, Taghi M Khoshgoftaar, Aaron N Richter, and Tawfiq Hasanin. 2015. A survey of open source tools for machine learning with big data in the Hadoop ecosystem. *Journal of Big Data* 2, 1 (2015), 24.
[9] Stephann Makri, Ann Blandford, Mel Woods, Sarah Sharples, and Deborah Maxwell. 2014. "Making my own luck": Serendipity strategies and how to support them in digital information environments. *Journal of the Association for Information Science and Technology* 65, 11 (2014), 2179–2194. https://doi.org/10.1002/asi.23200
[10] Daniel Moody. 2009. The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on software engineering* 35, 6 (2009), 756–779.
[11] Ivens Portugal, Paulo Alencar, and Donald Cowan. 2016. A preliminary survey on domain-specific languages for machine learning in big data. In *2016 IEEE International Conference on Software Science, Technology and Engineering (SWSTE)*. IEEE, 108–110.
[12] Carlton E Sapp. 2017. Preparing and architecting for machine learning. *Gartner Technical Professional Advice* (2017), 1–37.
[13] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*. 2503–2511.
[14] Juha-Pekka Tolvanen and Matti Rossi. 2003. MetaEdit+ defining and using domain-specific modeling languages and code generators. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. 92–93.
[15] Amy X Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. *arXiv preprint arXiv:2001.06684* (2020).

A APPENDIX - DEMO OUTLINE/SCREENSHOTS

A demo of the tool is available at https://youtu.be/YjOFt6XMuKA. Screenshots of the tool and examples of the diagrams created for a sample "property price prediction" project are shown in Figures 6-10.

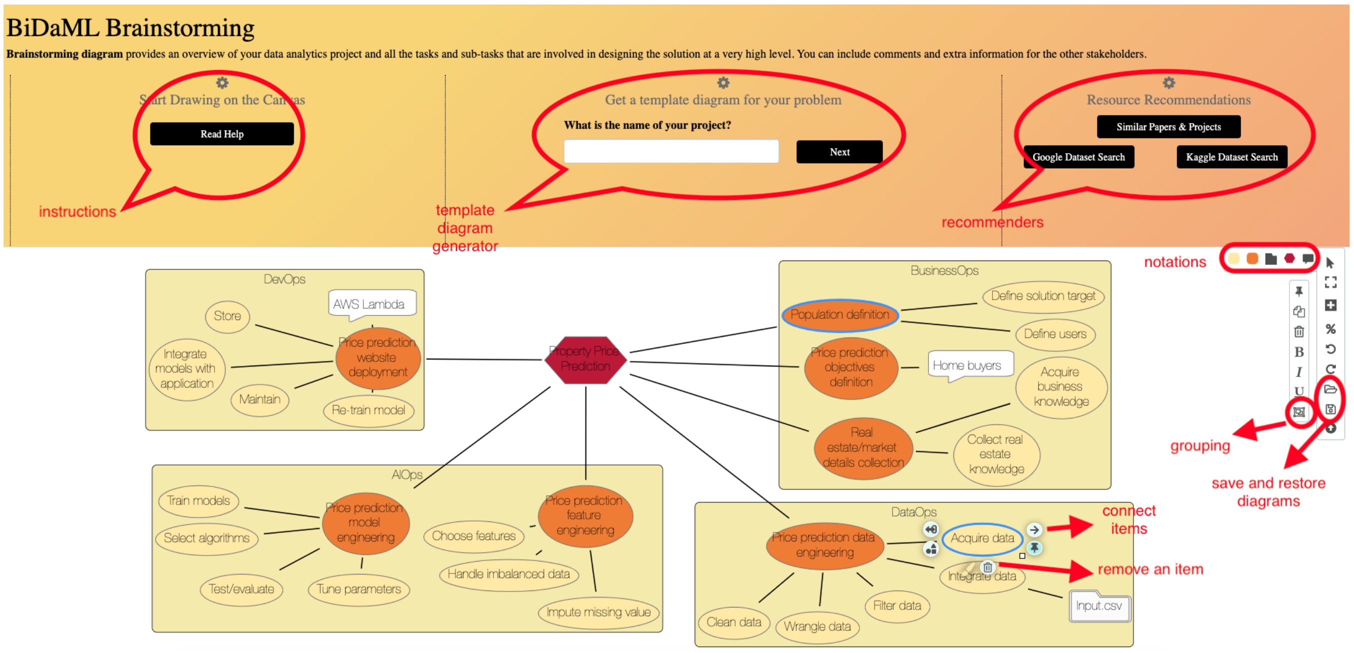

BiDaML Brainstorming
Brainstorming diagram provides an overview of your data analytics project and all the tasks and sub-tasks that are involved in designing the solution at a very high level. You can include comments and extra information for the other stakeholders.
Start Drawing on the Canvas
Read Help
instructions
Get a template diagram for your problem
What is the name of your project?
Next
template diagram generator
Resource Recommendations
Similar Papers & Projects
Google Dataset Search
Kaggle Dataset Search
recommenders
notations
grouping
save and restore diagrams
connect items
remove an item
DevOps
Store
AWS Lambda
Price prediction website deployment
Integrate models with application
Maintain
Re-train model
Property Price Prediction
BusinessOps
Define solution target
Population definition
Define users
Price prediction objectives definition
Home buyers
Acquire business knowledge
Real estate/market details collection
Collect real estate knowledge
AIOps
Train models
Price prediction model engineering
Price prediction feature engineering
Select algorithms
Choose features
Handle imbalanced data
Test/evaluate
Tune parameters
Impute missing value
DataOps
Price prediction data engineering
Acquire data
Integrate data
Input.csv
Filter data
Clean data
Wrangle data

Figure 6: The brainstorming diagram design page and an example created for a property price prediction project

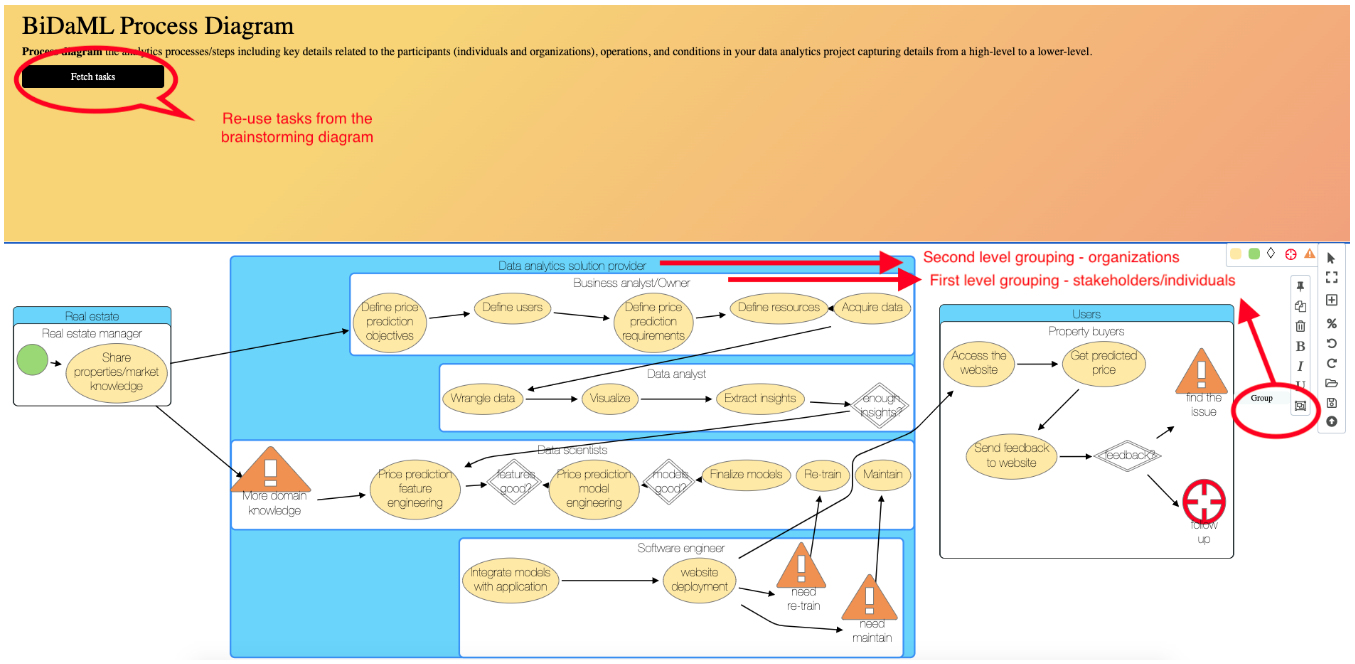
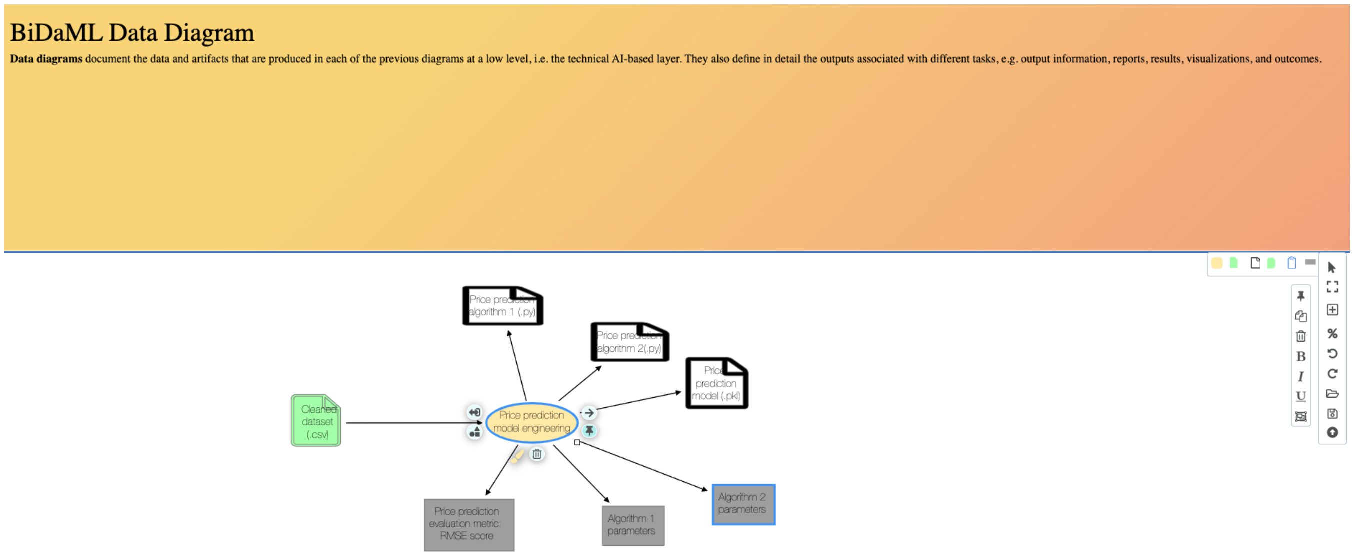

BiDaML Process Diagram
Process diagram the analytics processes/steps including key details related to the participants (individuals and organizations), operations, and conditions in your data analytics project capturing details from a high-level to a lower-level.
Fetch tasks
Re-use tasks from the brainstorming diagram
Second level grouping - organizations
First level grouping - stakeholders/individuals
Real estate
Real estate manager
Share properties/market knowledge
Data analytics solution provider
Business analyst/Owner
Define price prediction objectives
Define users
Define price prediction requirements
Define resources
Acquire data
Data analyst
Wrangle data
Visualize
Extract insights
enough insights?
Data scientists
More domain knowledge
Price prediction feature engineering
features good?
Price prediction model engineering
models good?
Finalize models
Re-train
Maintain
Software engineer
Integrate models with application
website deployment
need re-train
need maintain
Users
Property buyers
Access the website
Get predicted price
find the issue
Send feedback to website
feedback?
follow up
Group

Figure 7: The process diagram design page and an example created for a property price prediction project

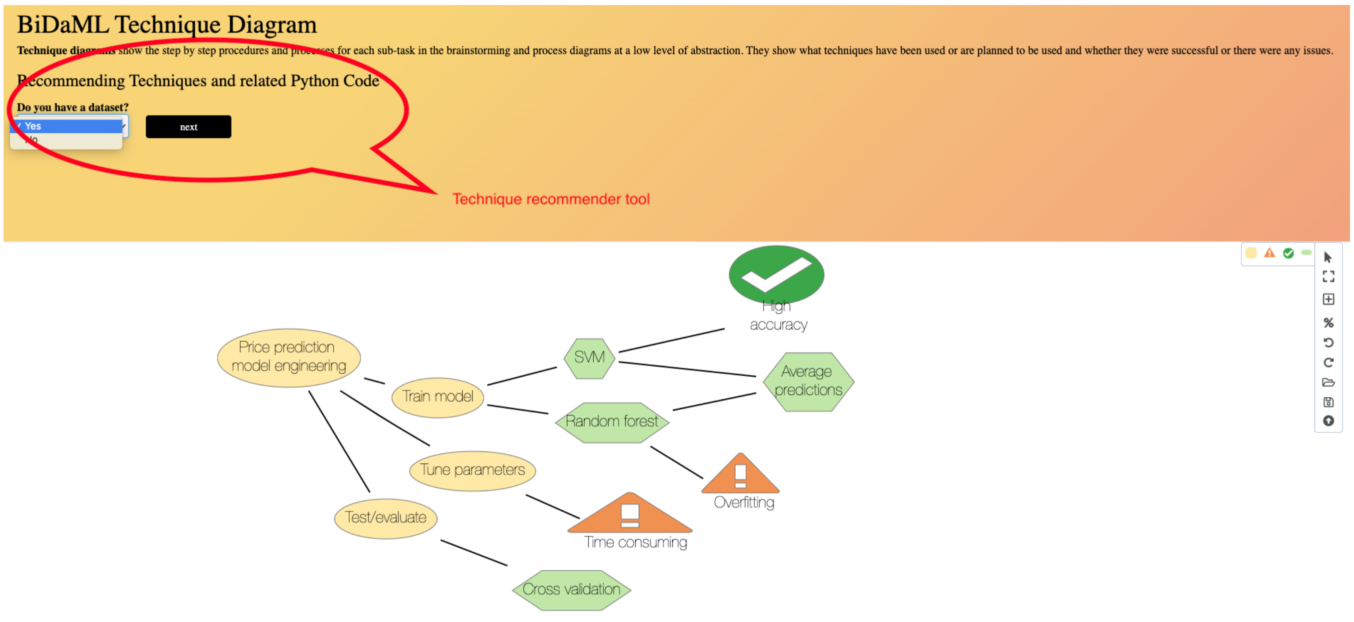


BiDaML Data Diagram

Data diagrams document the data and artifacts that are produced in each of the previous diagrams at a low level, i.e. the technical AI-based layer. They also define in detail the outputs associated with different tasks, e.g. output information, reports, results, visualizations, and outcomes.

Price prediction algorithm 1 (.py)
Price prediction algorithm 2(.py)
Price prediction model (.pkl)
Cleaned dataset (.csv)
Price prediction model engineering
Price prediction evaluation metric: RMSE score
Algorithm 1 parameters
Algorithm 2 parameters

Figure 8: The data diagram design page and an example created for a property price prediction project



BiDaML Technique Diagram

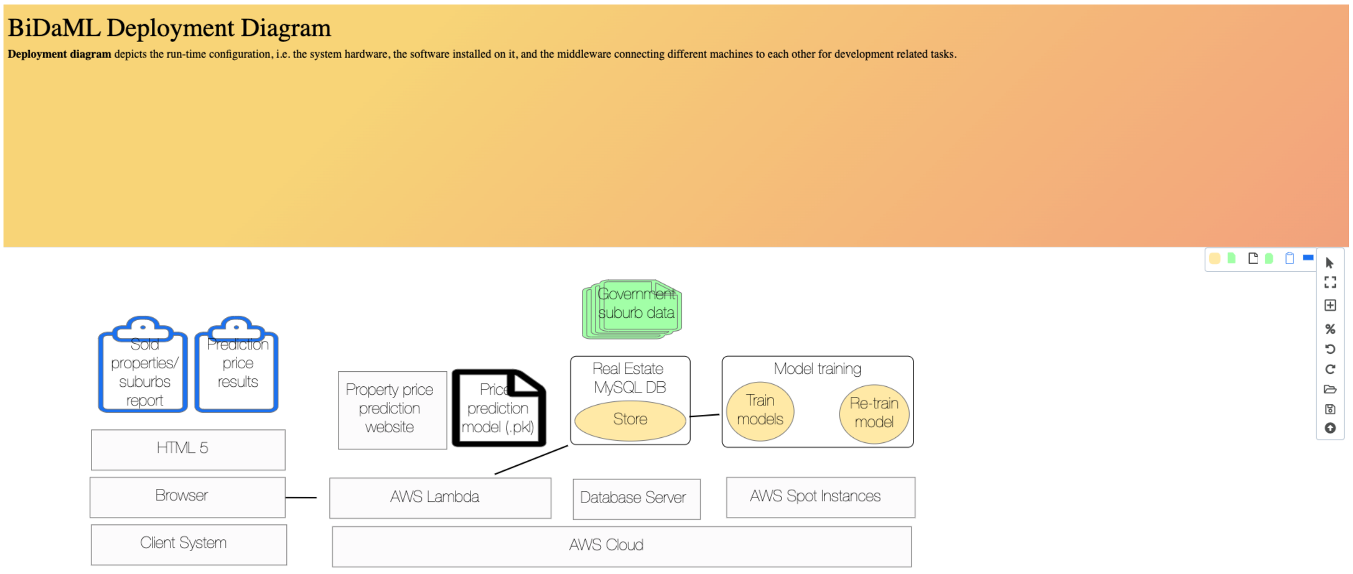
Technique diagrams show the step by step procedures and processes for each sub-task in the brainstorming and process diagrams at a low level of abstraction. They show what techniques have been used or are planned to be used and whether they were successful or there were any issues.

Recommending Techniques and related Python Code

Do you have a dataset?

[✓ Yes] No

Figure 9: The technique diagram design page and an example created for a property price prediction project



BiDaML Deployment Diagram

Deployment diagram depicts the run-time configuration, i.e. the system hardware, the software installed on it, and the middleware connecting different machines to each other for development related tasks.

Figure 10: The deployment diagram design page and an example created for a property price prediction project