Project - Methods

Data Transformation and Visualization in the ETL Process

Note to the instructors : Please begin grading the "Check-in #4" at Page 20.

Group # 24

Craig Groves
cgroves3@gatech.edu

Chamikara Dharmasena
cdharmasena3@gatech.edu

Jowanza R Joseph
jjoseph70@gatech.edu

Snejana Shegheva
sshegheva3@gatech.edu

Abstract. Mapping data from one form to another for its ease-of-use is at the core of the Extract, Transform and Load process. In this project, we design alternative prototypes for an internal interface of a transform task that prepares the data for use in a personalized recommendation system powered by Artificial Intelligence engines..

Problem Space

The data ingestion is described by the Extract, Transform and Load (ETL) process - a cycle that converts a raw data into structured records more convenient for further Data Analysis and/or uses for Machine Learning algorithms. Figure 1 shows the ETL process from the Source to the Destination. The entire cycle may be completely hidden from the user (full automation of data ingestion), or a human is required to guide the components of the process to reach their desired goal. In this project, we focus on the transform task that is centered around interactions with the user to alter the original data to meet their needs. For example, a user who looks at the weather feed in Fahrenheit may choose to convert it to Celsius.

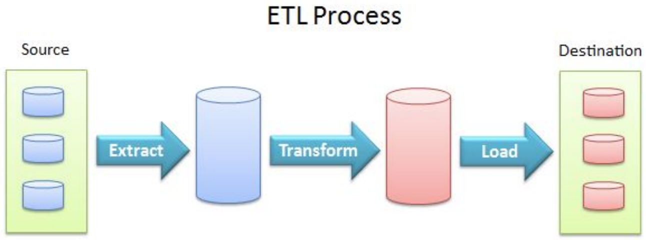
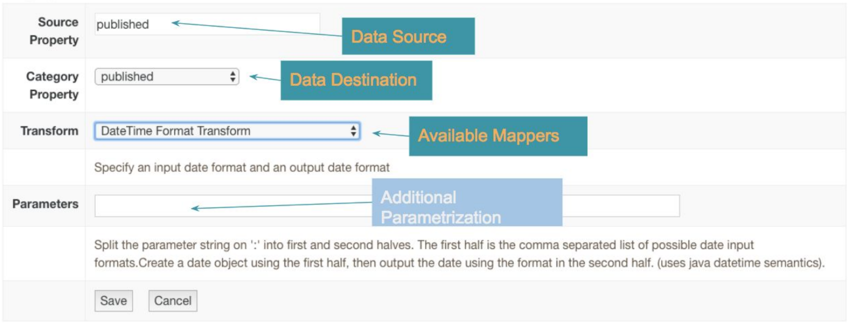


Figure 1: Extract-Transform-Load Process from Data Science Central on Open Source ETL tools (https://www.datasciencecentral.com/profiles/blogs/10-open-source-etl-tools)

To accomplish the data transformation task, a user needs access to the original data, as well as an arsenal of mapping tools suitable to the domain. Figure 2 shows an existing version of an internal interface[1] that we will be analyzing and redesigning to improve user experience in undertaking the transformation task.



Source Property	published
Category Property	published
Transform	DateTime Format Transform
	Specify an input date format and an output date format
Parameters	
	Split the parameter string on ':' into first and second halves. The first half is the comma separated list of possible date input formats.Create a date object using the first half, then output the date using the format in the second half. (uses java datetime semantics).

Figure 2: A version of an internal interface for transforming a single data field. Here, the data source is the data from the Arxiv Library collected via ATOM API. A user selects the original field, here the published date, and wishes to transform it to a different date format.

Our main goal is to assess all the weak areas of the existing interface in order to provide recommendations for alternative models that simplify the user interaction without assuming any pre-existing knowledge of the tool.

User Types

[1] A very rough version of a custom tool to perform user-driven ETL process.

Let's outline the user who is expected to engage in the data transformation task described above. The target user's expertise ranges from a novice (an external business client without extensive technical background) to an expert (data analyst interacting with the tool on a daily basis). The task assumes a basic proficiency with computer interfaces, which is a reasonable assumption given that the expected audience is professionals that use data to make business decisions. The task for different categories of users remains the same; however, the approach for accomplishing it may differ significantly. On the one hand, a user with substantial knowledge of the data (a domain expert) may want to explore complex relationships between the existing variables and generate new outcomes that capture the essence of the observed relationship. On the other hand, a user tasked with a data cleanup process, might not wish to delve into data intricacies and instead focus on the common exercise of standardizing the content (for example, change a separator in the numerical data).

A context of the task varies significantly and is mainly data-driven. A system for recommending movies involves data on user preferences and feedback. A method for optimizing financial portfolios streams data from financial markets and/or other sources. Therefore, an interface for accomplishing the described goals should be domain agnostic, and abstract the functionality of the transformation process to support a plethora of contexts. In general, a user needs access to a Web interface that allows uploading a data from many sources, and subsequently either loading it to a warehouse as is, or applying necessary and desired transformations to alter the original form. In order to accomplish a transformation task, users need to 1) know what data and in what form is currently available, 2) define the rules to convert the data from the original to the desired form, 3) see the final result to confirm the success, or receive a feedback for required adjustments in failure cases. The vast bulk of the user interaction with the interface is expected around the sub-task of defining a transformation rule. Therefore, a good design should focus on simplifying this step by providing an intuitive interface for data manipulation.

Data Inventory

Who are the users?

The users are individuals from a variety of backgrounds engaged in the data transformation task. These users can range from those whose primary role involves simpler tasks like data entry to tasks involving the extraction of relationships. Users are predominantly working professional ranging in their expertise from novice to experts without preference to age or gender.

Where are the users?

The users are primarily desktop and laptop users in an office setting. Being in an office environment, there are usually limited distractions which allow for more dedicated focus. They may also work remotely occasionally where the environment is a little more unpredictable. Working from home may be just as focused as in the office, or may have increased distractions.

What is the context of the task?

The context of the task is the ETL process. While performing the data transformation, other aspects in an office environment may vie for the user's attention. Email notifications, instant messages or telephone calls can trigger context-switching causing an increased cognitive load for the user. Also, the transformation task might have all of the information necessary to complete the load task, subsequent extractions and transformations may be necessary to finalize loading the data to its destination.

What are their goals?

Depending on the users' roles, the primary goals may vary. For the Data Engineering intense role, the goals are to get the data from its source loaded into its final destination completely and correctly. For Data Scientists the goal is to extract relationships via data transformations, which is essentially *understanding* the data. For Business Analysis, the goal is to build insights the support decision making in the business use cases.

What do they need?

They need the proper tools to perform ETL tasks. It must be determined whether or not the data currently exists in the destination to see if it still needs to be added. Users also need to understand the types of data coming from the source and what form it needs to be in for its destination. The difference between these two will dictate the kinds of transformations that the users will perform on the data.

What are their tasks?

The users' tasks are to determine whether the data from the source already exists in the destination. If not, users will need to determine the transformations necessary to prepare the data for loading into its destination. Once loaded, users will need to confirm that the data has been loaded correctly.

What are their subtasks?

After determining the data is not in the destination, the users' subtasks include gathering all missing data, locating the sources for that data and completing the transformations for the data loading task. If the transformations are complex, they may require the user to perform research or involve other team members before performing them on the data. Depending on the volume and types of data being added, the users may also need additional team members to confirm the correctness of the data after loading.

Needfinding Plan 1: Naturalistic Observation

Observation Details

For our naturalistic observation, we want to observe three people using ETL tools to complete the tasks and subtasks laid out in the data inventory. Since ETL takes place primarily on a computer and sometimes involves writing code, we need to be able to look over users shoulders, and thus we will ask their permission. Since we’re interested in the end-to-end processing of using the ETL tool, we want to arrange a time to “shadow” the participant. Our goal is to be minimally intrusive, we don’t want the user to change their behavior because we are watching. We will take detailed notes on what the user is doing, how they are navigating the interface and what the outcomes of their actions are. At the conclusion of this observation, we will take our notes back to the group, evaluate them and extract the user needs from them.

Biases

Observer Bias

Since our technique is that of observation, we have to be careful about our interpretation of that observation. It would be easy to interpret participants behavior in a way that confirms our hypothesis or to overemphasize confirming behavior while downplaying disconfirming behavior. To reduce this bias we should look to collect as much data as possible in the most granular form from the observation and then have an analysis on the observation afterward.

Needfinding Plan 2: Interviews

Interview Technique

Interviewees are persons recruited from work from two people in our group. They are people who meet the criteria in our data inventory. We will interview five people targeting different roles, and the interviews will take place in a quiet and secluded place like a conference room. To reduce bias, we will explain that we are researching the use of ETL tools in general, and avoid revealing our hypothesis to those who volunteer. We anticipate it will require 15-30 minutes per interview.

During the interview, the interviewer will take detailed notes for each answer and ask the follow-up questions if appropriate. At the conclusion of all interviews, the team (all four of us) will take the raw survey results, analyze them and record the users' needs.

Interview Questions

1. What are the ETL tools that you have used in the past?
2. What tools do you find easy to work with? (briefly explain what makes them easier to use)
3. What tools do you find hard to work with? (briefly explain what makes them harder to use)
4. What ETL tools do you currently use?
5. What data sources do you connect to most?
6. Are you satisfied with the performance of the tool (currently in use) when reading and writing data? (briefly explain the reason for yes/no)
7. Do you use the tool to clean/ reshape the data before transformations?

    a. If yes to (7), do you find the process easier within the tool or outside of the tool? (briefly explain)

8. Do you find the process of applying transformations to data easier? (briefly explain)
9. What are the transformations that you use frequently?
10. Are those transformations can be accessed quickly and easily?
11. Would you like to have suggestions on what transformations to be used?
12. How often do you find yourself redoing the same transformation due to errors?
13. Do you repeat some ETL workflows often (same transformations, different data)
14. What would you like to see changed about applying transformations on data?
15. Do you receive feedback from the tool after applying transformations? (failure/success of extracting and transforming data, other errors/warnings)
    a. If yes to (15), Do you find the feedback easier to interpret?

Biases

Confirmation Bias

Since we're starting out with the assumption that there is a flaw with current ETL tools, it is a natural human response to try to gather evidence that supports our views. We could easily extract information from respondents and interpret it as evidence for our claim about ETL tools. To reduce this bias, we should take as granular notes as possible during the interview process. Simply summarizing an interviewee is too dangerous as we may summarizing in a bias confirming way. We should instead come back to the interview notes or transcripts later and interpret them. Additionally, we should have our questions reviewed by an independent party to see if there is any bias in the questions themselves.

Voluntary Response Bias

Our participant pool is co-workers and people who would potentially have an interest in our well being, they may try to predict what the desirable answer is and respond with that rather than their true feelings. This would bias the results in creating a problem where one may not exist. In order to avoid this, we should say nothing of our hypothesis to the participants. They should know roughly the kinds of questions we're asking, but not necessarily our motivation for asking them. This will make guessing our motivations unlikely and reduces the potential for bias.

Needfinding Execution

1. Interviews

We have performed two rounds of interviews - one at the earlier stages that went in depth for how the current internal interface is being used, and the second at the later stage that did not limit the focus to the internal tool allowing a more general assessment of the task.

Summary of the results based on the interview responses.

The raw responses are recorded in the Appendix section A (including both rounds).

Who is the user	❏ Users working with clients (Product Leads); ❏ Data Scientists and Data Engineers; ❏ Software Architect who need to integrate the tools across all projects; ❏ UX designers who are tasked with implementing an interface to support data transformations
Where is the user	Mostly office environment with an easy access to the tools and data warehouse
What is the context	Typically a quiet environment but it may be interrupted by communication with co-workers
What are the goals	To understand the context of the loaded data (batches or in streams): ❏ What it contains ❏ How clean is it ❏ What are the possible relationships between data sources
What do they need	Depending on the scope of the project, and how well the

	data is structured users need a straightforward way to alter the data streams.
What are the tasks	The task will likely depend on what type of data is being fed to the system, but in general, the tasks include: ❏ Obtain the summary of the data ❏ Identify areas that require transformation ❏ Plan for applying the transformation: manual or automated
What are the subtasks	Once the users have established data sources that require transformations, they need to: ❏ Pinpoint the variable to undergo a transformation ❏ Link a Transformation function ❏ Apply the relation ❏ Confirm the result

Analysis of the results

➔ The workflow of the interface is not very intuitive, and it lacks *wizard-like* features to walk the user through the setup. Experiences users were able to navigate through the interface relatively easily, while novices struggled significantly with even the simplest subtasks, such as identifying how to Edit a variable.

➔ The interface provides only a partial set of operations needs to efficiently execute the task. As the interface does not provide an access to the data, it is hard to define what transformations need to be applied. Current users query the database directly to plan for the transformation tasks.

➔ The interface does not follow a standard design (button positions) that leads to confusion on the sequence of steps. For example, the interface currently has a great way to provide immediate feedback by letting the user *try* the transformation; however, because the designated location of this functionality is *below* the *Save* button, it is not clear if a selected transformation is *Applied* to the entire set.

➔ Generalization vs. Flexibility. Re-designing an interface must consider the trade-offs between how much flexibility it has (addressing the needs for very specialized audiences) and how many different users it can serve. For example, providing tasks for cleaning data (ex, extra spaces, date formats, numerical formats, etc) is applicable to most use cases, thus the interface must prioritize availability of basic units of transformations over the more complex tasks.

Bias mitigation

After the interview responses were collected from individual conversations, the results were circulated amongst the entire team to initiate a discussion of the outcomes. Since we targeted users from very different roles, a shared discussion placed the identified interface limitations in perspective - did *all* users agree on the priority of the limitation, and on the need to extend the functionality. For example, a Data Scientist argued for flexibility, while the Architect evangelized a common structure. By holding the debate on the interview results, we reduced the effect of the confirmation bias by allowing diverse interpretations from all interested parties.

Other biases, such as Voluntary Response and Social Desirability are lessened by excluding all potential solutions and interface re-design from the discussion. This allowed the users to explore their opinions and provide a deeper insight into the ultimate goal.

2. Naturalistic Observations

We conducted three sessions of naturalistic observations, in order to help meet the requirements of our needfinding. Notes from these observations can be found in Appendix B.

Summary of the results based on the observation notes

Who is the user	❏ Data Analysts (2) ❏ Data Engineer (1)
Where is the user	At their desks in an office environment
What is the context	Typically a quiet environment but it may be interrupted

	by communication with co-workers
What are the goals	To meet some requirements based on JIRA tickets they receive, this may include: ❑ Transforming data from a daily to a weekly granularity ❑ Cleansing data ❑ Publishing data
What do they need	They need access to the data sources, the requirements of the project and access to their ETL tools.
What are the tasks	The tasks are: ❑ Analyzing the requirements ❑ Creating the transformations ❑ Validating the transformations
What are the subtasks	The subtasks are: ❑ Reading the ETL tool documentation ❑ Contacting stakeholders to ensure comprehension of requirements ❑ Write SQL queries or DSL code ❑ Validate results

Analysis of the results

- Despite being marketed as “plug and play” tools, ETL tools observed still required some level of expertise with Structured Query Language (SQL).
- Participants needed to reference the documentation for their tools several times per usage
- Participants learned an additional Domain Specific Language (DSL) in addition to SQL
- Participants spent some cycles doing trial and error for their transformations before considering it complete
- Participants did not have a comprehensive way to validate that their transformations were correct

Bias mitigation

The main bias of concern for this observation was that of observer bias. We were concerned that conducting an observation, we'd be more likely to see the actions through our own biases about the interface rather than see the users point of view. To combat this, we focused our naturalistic observation on note-taking and did not ask the participants any questions. Additionally, we separated the analysis of the data from the collection of the data to reduce any bias.

Defining Requirements

Based on the results from the performed needfinding analysis, we can start describing the desired set of requirements for the interface that supports a data transformation task as part of the ETL process. The recommendations are provided for three areas - functionality, usability and learnability. Our analysis of existing users suggests that we need to cater to both, novices, and experts. The latter group benefits from the speed and efficiency, while the former is comforted with the explorative feel of the interface.

Functionality - a range of tasks supported for data transformation.

- ➔ The interface must let the user **select** a source data intended for modification, **choose** a transformation, **save** the configuration and **apply** the task to the dataset.
- ➔ The interface should give the user a **preview** of the outcome for the selected transformation **before** the task is applied.
- ➔ The interface should allow the user to create/edit/delete transformations, and optionally export it to a human-readable form.
- ➔ The interface should save the transformation flow for later use.

Usability - the quality of the available functionality.

- ➔ The interface must support a transformation that is based on multiple sources, for example, combining two fields into one, or performing mathematical operations on them.

- The interface must provide a clear distinction between actions for **saving** a configuration for a transformation, or **applying** the transformation the data.
- The interface must have the flexibility of changing the transformation settings and parameters easily at any level of the ETL process.
- The interface must provide a preview of data at user request. (At each step of the transformation)

Learnability - ease and speed for the task of transforming data.

- The interface should have a **toolbox of standard transformations** with meaningful names, and built-in tooltip that expands the capabilities of each transformation with examples.
- The interface should provide information on available datasets.
- Standard transformations should be categorized in order to improve the discoverability.
- The interface should provide **clear messaging** and notifications on the system's progress. For example, what is the most recent user's request, and what is its status.
- An interface should provide warnings of possible failures/warnings before the transformations.
- The interface should allow the user to change the preferences on the order of the available transformations. The current order is alphabetical, and it does not help locate the needed transformation quickly. **Options for ordering** may include:
    - Sort by relevance. Can the interface autodetect the transformations most compatible with the selected data type?
    - Sort by recency. Show the transformations on top which were accessed last.
    - Sort by popularity. A user may have a subset of transformations they access the most frequently, so it may be convenient to show a different set of transformations on top without requiring the user to scroll through the large list.

Since the interface described in this project is an internal tool, we have access to the user's feedback that we could use to evaluate the successes of the prototype. For each of the items outlined, we are planning to gathers three data points: 1) priority 2) estimate for implementation 3) feasibility and alignment with the team's goals.

Design Alternatives

Brainstorming

The individual brainstorming sessions are initiated with writing down the core problem - data transformation task - and allocating approximately one hour to a session on blackboard or a piece of paper. Each individual is instructed to come up with a list of at least ten solutions before they start analyzing and trimming the list. Each member of the team is encouraged to suggest alternative ways to think about the problem and not constrain themselves on the task representation of the existing interface. *Appendix* section *C* lists the raw results of the brainstorming sessions.

The group brainstorming took place over a Google Hangout session where we shared some interfaces for inspiration of simplicity and flexibility. A few interfaces that we looked at the Looker[2] product and ExplosionAI[3] demo services, where the latter does not entirely related to a transformation task, although it has features that are very attractive for data explorations tasks.

Selection Criteria

In the process of choosing the top three ideas, we defined the following selection criteria that reflects the requirements created as a result of a needfinding plan:

Criteria	**Description**
Customer Oriented	The idea should reflect the customer's needs, both internal and external and should not be designed for a single user.

[2] https://looker.com/guide/getting-started - a Business Intelligence Platform for performing ETL tasks.

[3] https://explosion.ai/demos/ - Demos for using Spacy for visualizing explorations of AI technologies on the example of NLP (Natural Language Processing) tasks.

Feasible in the implementation efforts	The idea should be feasible in the short term. After the design is approved, it should not require more than two engineers to implement the approach in two sprints (almost a full month).
Feature Coverage	The idea should cover the significant count of items from the requirements list.

Based on the selection criteria, these are the top ideas that we shall move forward with the prototyping:

- ➔ An interface that provides a single screen to view and transform one variable a time. The task is a good candidate to the basic form of wire-frames that can guide the user for through the transformation task by keeping a data sample visible at all times.
- ➔ An GUI interface that requires no domain specific language to build and create transformation. This provides a rapid learning curve with relatively little experience and in addition keeps the interface simple and organized.
- ➔ An interface leveraging direct manipulation allowing the user to control the flow of data through each transformation. It resembles the Query Designer view in SQL Server, which should be familiar to those users as well as Microsoft Access users.

Prototyping

Card Prototype 1

In this prototype we totally obfuscate domain specific language (DSL) attributes of typical ETL platforms and replace it with a drag and drop interface. This design aims to make the interface more direct by representing the columns in the table as selectable icons users and drag and drop and create new tables and visualizations with. As seen in Figure 3, users can select from a list of available facts (quantitative data) and dimensions (variables) to create new transformations with. They can then mix and match that data with a number of preset transformation options. Users are not allowed to write any custom DSL, but can accomplish everything they wish in the DSL with the interface.

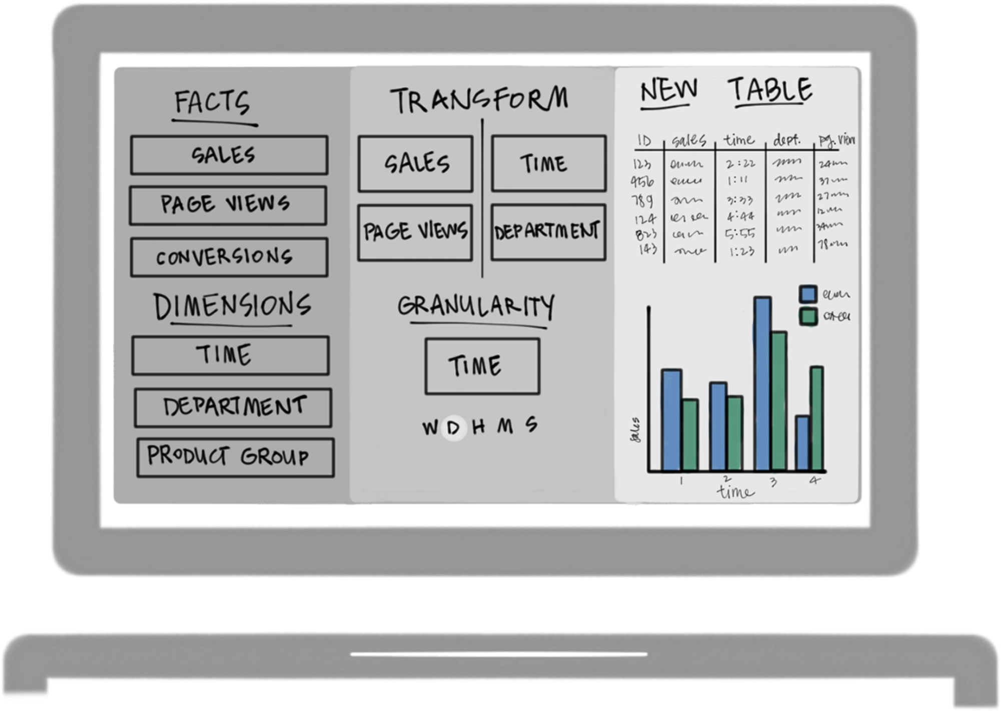


Figure 3. A Drag & Drop Card Prototype for Transformations

The key features of this interface are as follows:

- ➔ All transformations are completed by drag & drop removing the friction of using a DSL.
- ➔ The design enables the user to feel safe to explore actions can be undone by simply changing the facts and dimensions in the transformation area.
- ➔ The interface focuses on simplicity, by narrowing the focus of what users can do and eliminating any superfluous features.

Wireframe Prototype 2

Inspired by the Query Design view in SQL Server, the third wireframe prototype shown in Figure 4 features multiple nodes, depicted as dropdowns, for representing data locations (source or destination) or transformation.

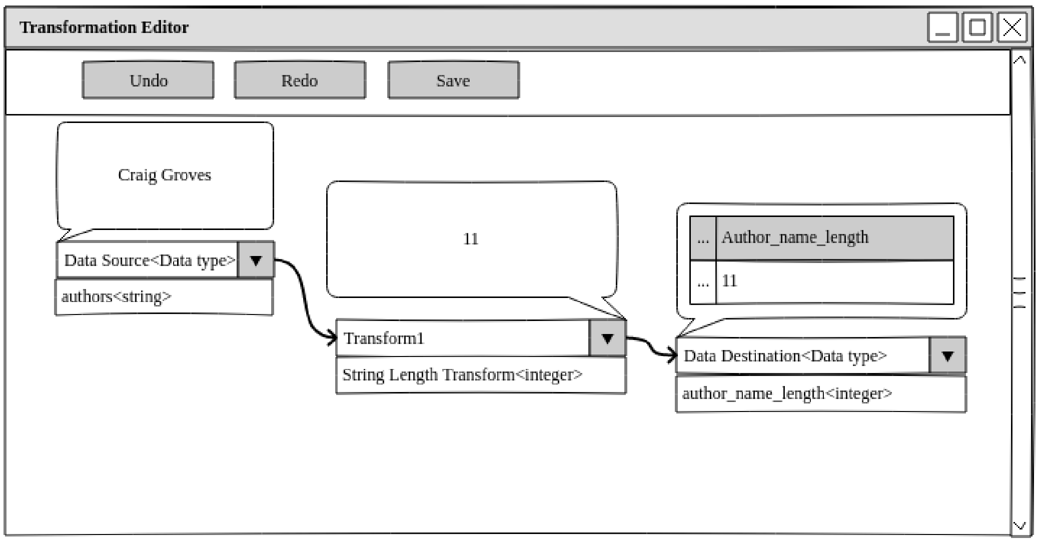


Figure 4. Query-designer style interface prototype

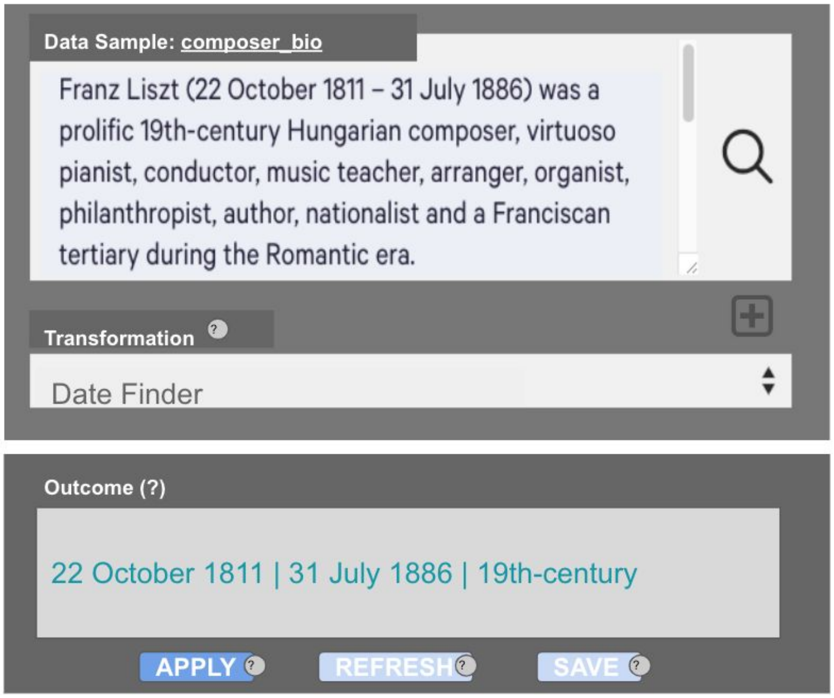
Leveraging the principles of direct manipulation users can control the flow of data through various transformations to its ultimate destination.

- ➔ The drop down menus allow users to view all available data locations for sources and destinations or data transformations.
- ➔ The dropdowns affords typing which allows the user to search for a desired transformation or data location. Each dropdown features an autocomplete functionality for convenience to shorten the amount of time to select an option.
- ➔ And, finally, each dropdown includes a balloon that shows a preview of the data being selected or transformed. The data source balloon shows its original state. The transformation balloon and data destination balloon shows the result after each transformation and the final state of the data at its destination, respectively.

The described prototype includes a redo and undo buttons to perform those functions on the actions that the user has taken. Additionally there is a save button for saving the transformation that was created.

Wireframes Prototype 3

The task described in this project involves knowledge of data intended for transformations. Providing the user with a view that shows a sample of the data aligns well with the *perceptibility* principle that keeps the user informed about what is going on through appropriate feedback. Figure 5 demonstrates a prototype of the interface that start with a *view* of the data that user selected for a transformation (contrast this with the existing interface presented in the Figure 2). In the alternative interface, the user is presented with the data sample that they can search through thus making them feel that they are closer to the data. Right beneath the sample, the user can select a transformation from the drop-down list. And importantly, the new interface shifts the focus on the outcome that leads to a better experience for goals of exploring and learning the data.



Data Sample: composer_bio

Franz Liszt (22 October 1811 – 31 July 1886) was a prolific 19th-century Hungarian composer, virtuoso pianist, conductor, music teacher, arranger, organist, philanthropist, author, nationalist and a Franciscan tertiary during the Romantic era.

Transformation: Date Finder

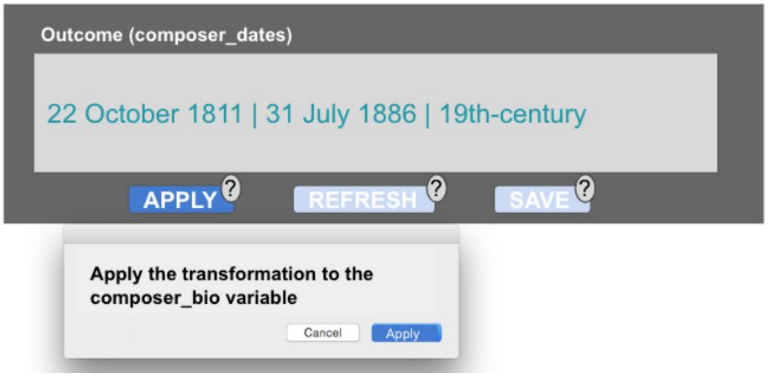
Outcome (?)

22 October 1811 | 31 July 1886 | 19th-century

Figure 5: An alternative interface for the task of Transforming a variable. The screen includes three sections: 1) Data View that provides a sample of the data, 2) a drop-down section from where the user can select a transformation function, and 3) the output example that demonstrates the outcome of the applied transformation.

Features of the prototype:

- The user selects a variable (ex, composer_bio) from the existing screen
- The user is immediately presented with a **sample** of the data
- The user can **search** through the data using the magnifier affordance
- The user can **select a transformation** from a drop-down menu that comes right under the sample
    - The tooltip with the question mark explains what "transformation" means
- The user can **add** another variable with the "plus" button affordance
    - This would support the case beyond current one-to-one transformation
- The user can **see** the outcome **before** applying the transformation
    - If the transformation is new, the outcome will be followed with a question mark in parenthesis to signal an unknown property
- The user can choose to **apply** the transformation



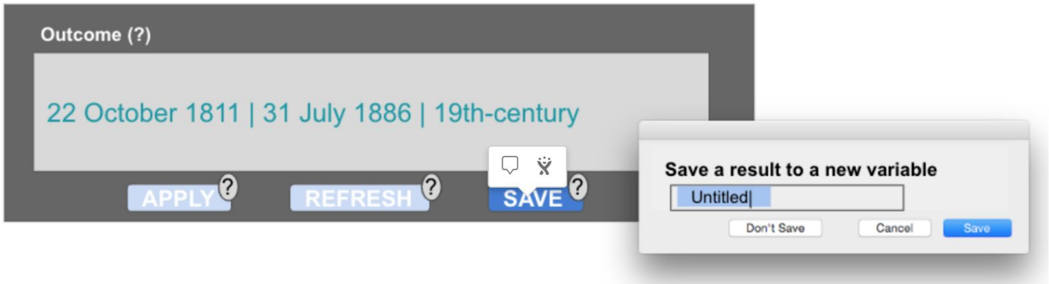
Outcome (composer_dates)

22 October 1811 | 31 July 1886 | 19th-century

Apply the transformation to the composer_bio variable

Figure 6: A section of the screen that allow the user to apply the selected transformation.

- The user can **save** the transformation where they are prompted with giving the new property a name.


Outcome (?)

22 October 1811 | 31 July 1886 | 19th-century

Save a result to a new variable

Untitled

Figure 7. A section of the screen that allow the user to save transformation with a new variable name.

- ➔ The user can **refresh** the transformation to re-do the transformation if they have mocked with the input
- ➔ And, finally, the interface supports transformations beyond one-to-one where the user can select multiple variables via “plus” sign and apply a transformation that acts on more than one variable.

Evaluation Planning

The strategy is to analyze at least two of our three suggested prototypes using Qualitative and Predictive evaluation approaches. All prototypes are intentionally distinct from each other in terms of targeted users and the spectrum of features. Therefore, our evaluation plans will give us a perspective on how the prototypes compare with regards to the coverage of requirements, and ease of use. Since the task selected for this project is carried out using an internal interface, we have access to users who can give us a direct feedback either via Interviews or Think-aloud studies. We will also include a Cognitive Walkthrough to further understand the user’s thought process especially focusing on the learnability aspect of each prototype. The crucial metric of this evaluation plan is to assess whether or not the user's needs have been met. A detailed list of *hits* and *misses*, in addition to the user's overall sentiment will serve as a decision rule on whether or not to advance the suggested prototype to the next stage.

Think-Aloud Study

For this step of evaluation, the plan is to engage a single user in the think-aloud study. The chosen user, Product Manager with a client-facing role, has significant experience with the task at hand and is expected to provide good feedback on the wireframe prototypes. The evaluation will take place in the work setting where the user will be shown an alternative interface to perform their usual task of data transformation. The results of the conversation will be summarized and recorded the exchange takes place. Any notes drafted during the discussion will be collected as artifacts of the study. The goal of this evaluation is to collect early feedback on the suggested idea and estimate user's interest in taking it to the next step of implementation.

To get started with the evaluation of each prototype, we will ask them to perform a specific transformation task while thinking aloud through the steps. The directions will be based on actual use cases to avoid biasing the questions that benefit the specific features of each prototype, We will not be providing detailed instructions on **how** to accomplish the goal.

- ➔ Question 1 - Field Parsing use case: Let's say you have collected data on the recently published papers in the field in Machine Learning and Artificial Intelligence. You have a field that contains a list of *keywords,* and let's assume you only want to get the first-mentioned keyword from the list. How would you accomplish this task in the new interface?
- ➔ Question 2 - Term Spling Use case: Let's say that in the same data set, you have a field that captures the *published* date. If you would like to split this into a *year* and *month* variable, how would you go about it?
- ➔ Question 3 - Time Series Use case: Let’s say that you have collected house pricing data over time. If you would like to view the trend of the data on different scales - weekly, monthly, annually, etc, what steps would you take?
- ➔ Question 4 - Use case: Let's say you have two *date* fields - *submission date* and *acceptance date.* Your goal is to create a variable that stores the elapsed time between the two dates. Do you think the new interface will sufficiently guide you through the process? What do you imagine the interface should do if

you are stuck on the task, such as - the recommendations for the transformations do not capture your intentions very well?

- ➔ Question 5 - General Satisfiability: What tasks in your opinion are more efficiently performed in the new interface when compared to the current version? How about less efficiently? Would you still prefer the current interface and why?
- ➔ Question 6 - Learnability: Do you expect a significant amount of guidance from the interface, and which areas? For example, would you be able to find the variables you need to change quickly? How quickly do you expect to see a transformation that suits your needs? Do you plan to request a different set of recommendations? Would you like to quickly access *all* available transformations?

Interviews

Interviews were conducted over the course of 2 hours in a conference room setting. 5 participants were selected at random from a pool of 10 potential interviewees. Participants were a mixture of Data Analysts, Data Engineers and Data Scientists. Each of them perform data transformations with software as part of their daily job. The goal of the interview is to assess how well our prototype (pictured in figure 3), helps the users accomplish their goals. In order to measure this qualitatively, we need to first establish the user's goals as a stated preference. Following this, we need to ascertain the context in which users try to accomplish these state goals. Finally, we'll ask about the participants about how they would perform common tasks in the interface, as well as their overall impressions of the interface. All questions for interviews can be found in the Appendix.

Evaluation Execution

Think-Aloud Study

The evaluation span across four sessions on two different days with the same participant. The chosen participant has significant experience with the task at hand that contributed to very productive session across all three prototypes.

The user's thought process across all cases can be summarized with the following steps:

- ➔ Find the input variable from a either a drop-down list or from the facts section
- ➔ See the list of recommendations in the transforms page or in the given drop-down list
- ➔ Apply the transformation and evaluate the results based on the observed output

The summary of the results across each prototype is recorded in the table below that allows us to compare the pros and cons based on the observations from user's ability to perform the task, and their subsequent feedback for each use case:

Questions	Card Prototype 1	Wireframe Prototype V1	Wireframe Prototype V2
Q # 1	The user had no difficulty identifying the input variable in the Facts section, however, they we unclear about how to use the Dimensions. Similarly, the Granularity in the Transform section seem to not apply to the extraction use case	The user performed a transformation for this use case in smooth flow without pausing to think about the next steps. User commented on the simplicity and the presence of immediate feedback over hover over (the speech bubbles in the interface)	Before initiating their thinking aloud process, the user paused for a few minutes to observe the new interface. After thinking about it for while, they talked through the transformation with ease. The positively commented on the ability to search through the given selection of of text.
Q # 2	This case is similar to the first one as it expects the extraction, except in this case the extraction one to many was not supported by the prototype	The user assumed the based on the selected transformation, the third block will automatically split into two variables to reflect the side effect of the transformation. The user recommended adding another screen that	Similarly, as in the second prototype, the user recommended adding another screen that exemplified different uses cases: - one to one - one to many - many to one

		exemplify this scenario.	
Q # 3	The user responded very positively to this use case, and it took them only a few seconds to navigate in the variable selection, transformation, and interpretation of the results	The user paused on this use case, and eventually asked a few clarifying questions. They did not immediately figure out how to apply a time series transformation. They commented that perhaps visualizing time series is outside the scope for the variable transformations.	The user correctly anticipated where the time series would be visualized. They however pointed out that they would sometimes like the ability to change the granularity similar to the prototype 1. They asked if the a transformation can be parameterized to accept granularity preferences
Q # 4	The prototype is limited to one-to-one transformations (which are the most common case in standard ETL). The use case provided in this study is a more advanced scenario of combining (or performing a mathematical operation on more than one field). After thinking about the interface for awhile, the user concluded that use case is not fully supported	Although, the prototype did not explicitly demonstrate the presence of the second variable, the user was able to infer and interpolate how one would add the second variable and apply a transform that is a function of selected two.	The user quickly identified the "plus" sign that add more variables to the transformation. They liked how this prototype was more explicit in directing the user for making other types of transformations.
Q # 5	The user really liked being able to see everything on one page, and especially being able to drag and drop the variable from the Facts stage to the subsequent transformation phases. The user mentioned	The user was satisfied with the simplicity of the prototype, and although, they did not find it very appealing visually, they agreed that for this fidelity of the prototype, it was sufficient to be presented	The user was satisfied with the interactiveness of the prototype (search through the text and editing the text to play with the transforms)

	resemblance to Splunk[4] mainly because of its appeal to analyze time series.	in a draft-like mode.	
Q # 6	The user commented on the learnability of the interface, and expected minimal guidance for the variables that required one to one transformation. The user pointed the limitation of flexibility for more advanced use cases, and felt that the prototype is restricting them to only a subset of tasks.	The user suggested to move the "Undo, Redo, Save" buttons to the bottom of the screen to be more consistent with other similar applications. Generally, the user responded very positively on the efforts required to make a transform. The concluded that it would be sufficient for most of the use cases.	The user really liked the look and feel of the interface and commented on its high fidelity (although nothing has been implemented yet, the user felt that they understood the process). The main feedback was around the buttons to Apply, Refresh and Save - they felt that some of it maybe redundant. The user responded very positively to the use of tooltips throughout the interface to guide the new users and encourage the exploration.

Interviews

To start the interview, participants were asked questions about their day-to-day workflow, and interviewers took liberties to ask follow up questions when necessary. Overall, participants spent 50% of their time performing some ETL related tasks (self reported). Of that 50% of time, they spend 90% of time procuring datasets and performing some kind of transformation on that data. Participants used a wide array of tools to accomplish their data procurement and transformation tasks including Microsoft Excel, Structured Query Language (SQL), Looker, and custom Python code. Participants stated their goals were to provide business value to their clients by analyzing data, specifically to find abnormalities or insights in the data to

[4] Splunk is a software for searching, monitoring and analyzing the large sets of data - https://www.splunk.com/

surface to their clients. Among their subtasks were finding ways to look at the data in non-conventional ways, and discovering time-based anomalies that may save their clients time or money. All participants said they primarily perform data procurement and transformation tasks from work, but on a rare occasion they may perform it from their laptops at home over VPN.

Participants were presented with a large copy of the interface before being asked about it. Participants were asked to create several new views and asked how they would go about performing that task. For completion of these goals, participants were largely (4 out of 5) able to complete the tasks without any additional assistance. The single participant that wasn't able to complete the task had a questions about how data would be joined and if they would have any control over the visualizations that were produced from the transformation. After clarifying these issues they were able to complete the task.

All 5 participants cited that the prototype was user friendly and aesthetically appealing. Open questions about how the interface could be improved produced suggestions that ranged from including more fine grained controls for transformations and data types to expressing how data was joined together. A summary of the interviews can be found in Appendix D.

GOMS Models

The card prototype(**Figure 3**) allows users to create transformations with a drag and drop approach. New tables and visualizations can be created without having to write any custom DSL. Wireframe prototype 2, which featured a Query Designer type view, was also evaluated using a GOMS model shown below. An estimate is provided for the amount of time taken to perform the operation listed where (s) and (m) are the number of seconds and minutes respectively.

Task Breakdown

- ➔ Open application
- ➔ Select table/tables from facts/dimensions list.(Left Pane)
- ➔ Drag and drop selected table/tables on to the desired transformation task
- ➔ The right pane will display the results (table, charts)

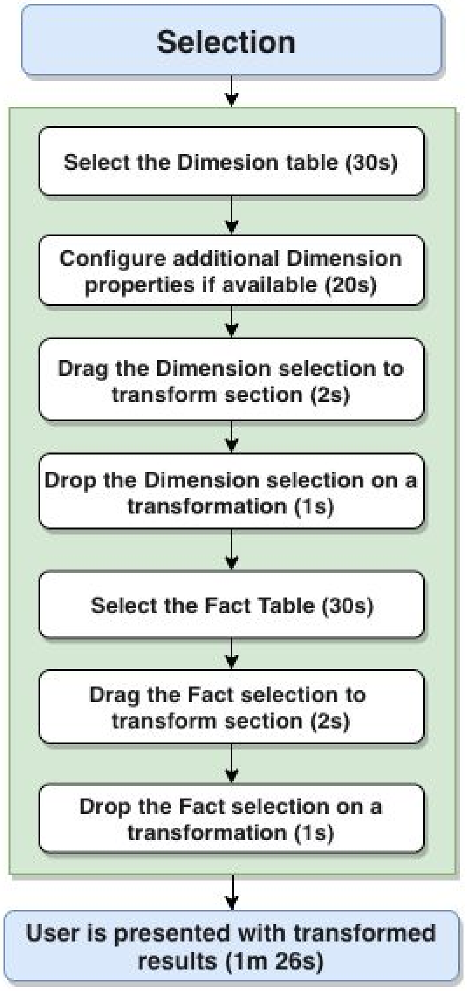

Selection
Select the Dimesion table (30s)
Configure additional Dimension properties if available (20s)
Drag the Dimension selection to transform section (2s)
Drop the Dimension selection on a transformation (1s)
Select the Fact Table (30s)
Drag the Fact selection to transform section (2s)
Drop the Fact selection on a transformation (1s)
User is presented with transformed results (1m 26s)

Figure 8. The overall GOMS model for the Card Prototype.

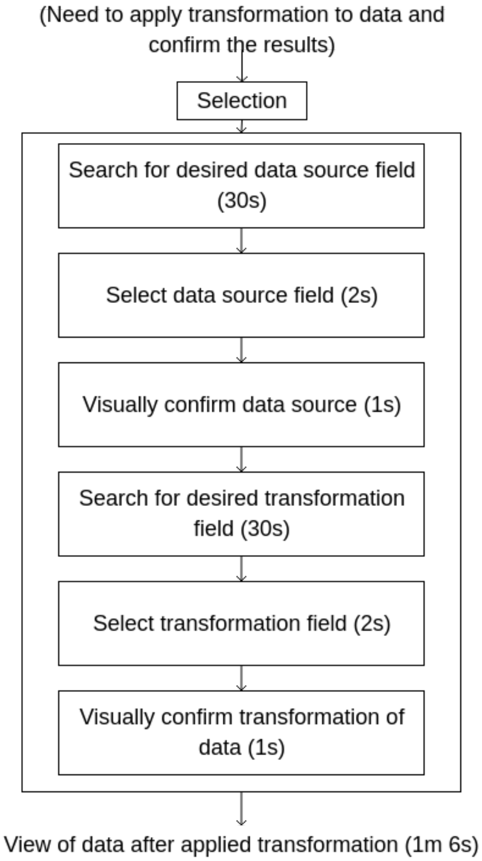

(Need to apply transformation to data and confirm the results)
Selection
Search for desired data source field (30s)
Select data source field (2s)
Visually confirm data source (1s)
Search for desired transformation field (30s)
Select transformation field (2s)
Visually confirm transformation of data (1s)
View of data after applied transformation (1m 6s)

Figure 9. Wireframe Prototype 2 GOMS model

The Wireframe prototype 3 (**Figure 5**) users to apply transformations quickly and easily. A GOMS model is used to evaluate the prototype.

Goal: Apply transformations to a data set.

Goal: Open a dataset. (A preview of a sample dataset is provided)

```
            -Type a keyword to search a dataset.
            -Press "Enter" or click the magnifier icon.
Goal: Select a transformation
           - Click on the drop-down icon.
           - Select the desired transformation.
                Goal:  Add another variable.
                        -  Click "+" button.
                        -  Select the desired variable.
Goal: Apply the transformation
            - Click the "APPLY" button
                Goal: Save the transformation.
                        - Click the "SAVE" button.
                        - Enter a property name.
                        - Click the "Save" button.
                Goal: Re-do the transformation.
                        - Click the "REFRESH" button.
```

Wireframe prototype 2, which featured a Query Designer type view, was also evaluated using a GOMS model shown below. An estimate is provided for the amount of time taken to perform the operation listed.

References

1. Vassiliadis, Panos. "A survey of extract–transform–load technology." *International Journal of Data Warehousing and Mining (IJDWM)* 5.3 (2009): 1-27.
2. Beaudouin-Lafon, Michel, and Wendy Mackay. "Prototyping tools and techniques." *Human Computer Interaction-Development Process* (2003): 122-142.
3. Nielsen, Jakob. "Usability inspection methods." *Conference companion on Human factors in computing systems*. ACM, 1994.

Appendix

A

Interview Questions from Internal Team

Section 1 - Data Presentation

When deciding to transform a property, what is the value for you in being able to see the sample of existing data for the given property? If you do not see a value on that, please explain why. If you see a value, please provide a recommendation on what constitutes a good sample, for example, you want to see the most recent values, the most frequent values, the most extreme values, etc. If you choose to say "yes" to the value, then please justify your needs. Here, you can describe your role, and why is this important to you.

Section 2 - Data Transformation

How frequently the data you loaded requires additional manipulation? What are your typical scenarios for transformations? When transforming a variable, do you need access to the other variables in a single record? What is one type of transformation that you need/want the most?

Section 3 - Feedback/Recovery

Do you feel that the current system’s feedback is adequate? For example, recall scenarios where you wanted to transform a variable, but were struggling with the interface in terms of selecting needed information (selecting a transform type, configuring the transform, etc.). What is your experience at the times where the transformation you applied had an unexpected effect? Did you have enough information to understand the issue and/or recover your data and re-iterate the process?

Interview Answers from Internal Team

Interview Raw Responses

Role: Senior Product Manager, Client Lead

Question 1 - Data Presentation

•I end up going through this process for at least half of our customers

•Data presentation is helpful, but only critical when a more complicated transformation is required (e.g., regex). Typically, I will look at the data PRIOR to this screen because I need to make the decision to do the transformation before I land here.

•Once on this screen, it would be helpful to see the 'would be' output before it's changed.

• NOTE: this screen is currently just a config, not a processing screen. A more

seamless flow would be very helpful!

Question 2 - Data Transformation

•I tend to think that an Excel/CSV/DB table format is the easiest way to look at

this data (horizontally, one record per row). This is the way we load data into the

system initially, too, so I'm already familiar with looking at data in this format.

•Probably my most frequent transforms at this point are the concatenation (combining two or more values) or trimming/parsing/stripping from fields (e.g., removing characters)

•I don't do bucket transforms here, probably because I find them too difficult/not

intuitive

•The dropdown has MANY transforms I have never used... largely because I don't

know what they do.

•The parameters field is applied inconsistently and isn't clear. What is supposed to go there? How do we configure these params? Better documentation needed.

•On my wishlist - it would be great to configure transcendence/standardization at this point too. Transcendence and tagging both feel like transforms to me.

<u>Question 3 - Feedback/Recovery</u>

•I think the issue here is two-fold. 1. This is just a configuration screen. It doesn't kick off any processing, so you could set this and never see the outcome. 2. We don't have an easy way to see the processed data until it is moved to the platform (aka production). Exposing these in a staging area would be ideal.

• There are additional steps required to get the data wired all the way through to 'final output' (e.g., the partner property).

•Correcting errors is especially hard with our data processing tool. I believe we update a transform and reprocess to the correct values, but I'm not sure if that catches everything or if old/bad data can remain (e.g., if you mistakenly use a transform that in one case yields a value for a source entity... if you correct it, so now that new source entity has no value, does the old value go away?

<u>Role: Chief Technical Officer</u>

<u>Question 1 - Data Presentation</u>

•Not using mental energy to imagine how data looks like before the transformation is very important to me; This allows to focus on the interface. Plus, imagination could be exhausting :)

<u>Question 2 - Data Transformation</u>

•I am involved in rapid prototyping for a variety of clients. During this phase, the data we get is very unclean; therefore I have to perform a lot of transformations.

•Main areas of transformations: Dictionary Mappings, Combining fields, Cleaning Up.

Question 3 - Feedback/Recovery

•Try it functionality we currently have is a great way to provide feedback in the form of almost interactively reshaping the data

• An area where I stumble is around naming for the current transforms. A name is not always representative.

Role: Lead UX Designer

Question 1 - Data Presentation

•It would be great to have a sneak peek into the data; however, one should be careful about not distracting the user from the task. Essentially, if the action of taking a look at the data underneath requires selecting multiple options, it does not improve the user's experience overall.

Question 2 - Data Transformation

•Although, I am not currently using this feature, I can imagine a need for a transformation that requires multiple variables.

Question 3 - Feedback/Recovery

•The main advice in providing good feedback to the user is for the interface components to be consistent with the standard interfaces (buttons especially). For

example, the actions should be ordered as Cancel, Apply, and Save. In our case, Try it button is below Save which might confuse the user.

Role: Software Architect

Note: For this interview, I did not have a chance to record the answers immediately, so I paraphrased their main takeaways

Question 1 - Data Presentation

•Although it is a noble goal to show the user a sample of the data, scalability of

such requests need to be considered carefully. Would this additional feature,

although nice, interfere with the overall task.

Question 2 - Data Transformation

•Too many transformations can be generalized and combined into a single function. No need to be too granular in what each function can do.

Question 3 - Feedback/Recovery

•The users currently may suffer from system's frugal messaging when something goes wrong, and its verbosity for the typical workflow

Role: Data Scientist

1. What are the ETL tools that you have used in the past?
    a. Various versions of home-grown tools across multiple companies. Interfaces on top of ETL involving programmatic streaming, spark etc. using parsers and lambda functions. Product managers unable to complete transformations due to non-GUI transformations. In the previous companies, we have applied ETL to only one domain where the data structure for the input is fixed which allowed for a programmatic use.
2. What tools do you find easy to work with? (briefly explain what makes them easier to use)
    a. My criteria for easy to use is requiring the least amount of steps required to complete the task.
3. What tools do you find hard to work with? (briefly explain what makes them harder to use)
    a. Tools that hide that data are hard to work with, especially for Data Scientists, who need to make the transformations based on how the data looks like. So seeing the data is important.
4. What ETL tools do you currently use?
    a. An internal tool developed within a small team using databases and a few JavaScript screens.
5. What data sources do you connect to most?
    a. Databases, S3 (Amazon), local files
6. Are you satisfied with the performance of the tool (currently in use) when reading and writing data? (briefly explain the reason for yes/no)

   a. No. The tool we currently use does not expose the data which makes it hard to use efficiently. A preview of the data would help to determine the right kind of transformation. It's better to get the transformation that has most coverage of the input data quickly. Also testing out the transformation on the data could take a long time depending on the amount of data.

7. Do you use the tool to clean/ reshape the data before transformations? If yes to (7), do you find the process easier within the tool or outside of the tool? (briefly explain)
   a. Yes, if time is of the essence, I perform data cleaning with Pandas in Jupyter Notebooks and then uploading the results. This is not always possible if we are dealing with production pipelines.
8. Do you find the process of applying transformations to data easier? (briefly explain)
   a. Same as data cleaning.
9. What are the transformations that you use frequently?
   a. Extract entities. Also less frequently, performing one-to-many, many-to one-transformations on the same piece of data.
10. Are those transformations can be accessed quickly and easily?
    a. Yes
11. Would you like to have suggestions on what transformations to be used?
    a. Yes, but I would also like to get an access to all possible transformations if the recommendations are not returning what I am looking for.
12. How often do you find yourself redoing the same transformation due to errors?
    a. Many times. Transformations might not completely transform all of the input data, so there is a lot of testing and refinement to get the transformation right.
13. Do you repeat some ETL workflows often (same transformations, different data)
    a. Yes
14. What would you like to see changed about applying transformations on data?

   a. Make it transparent by showing a sample of data, and giving a preview of the output.

15. Do you receive feedback from the tool after applying transformations? (failure/success of extracting and transforming data, other errors/warnings). If yes to (15), Do you find the feedback easier to interpret?
    a. No feedback is currently available. We have to look at the database to query for the results.

Role: ETL Developer

1. What are the ETL tools that you have used in the past?
    a. SSIS, SAS tools, Scala IDE
2. What tools do you find easy to work with? (briefly explain what makes them easier to use)
    a. They were not exactly easy to use. SAS Enterprise Guide was a decent tool
3. What tools do you find hard to work with? (briefly explain what makes them harder to use)
    a. Scala IDE, I have to write Scala code for all the transformations
4. What ETL tools do you currently use?
    a. Scala IDE (with Spark)
5. What data sources do you connect to most?
    a. HDFS, S3 ,MYSQL,Greenplum, MongoDB
6. Are you satisfied with the performance of the tool (currently in use) when reading and writing data? (briefly explain the reason for yes/no)
    a. Yes to some extent. Errors occur sometimes due to unavailability of source data or the destination
7. Do you use the tool to clean/ reshape the data before transformations? If yes to (7), do you find the process easier within the tool or outside of the tool? (briefly explain)
    a. I import data to my local memory in order to clean them and reprocess them.
8. Do you find the process of applying transformations to data easier? (briefly explain)

   a. I have a set of transformations and their Scala code (pre-written by me)ready to apply with some changes on parameters.

9. What are the transformations that you use frequently?
   a. Joins, data quality, data profiling, and extractions
10. Are those transformations can be accessed quickly and easily?
   a. It takes some time to set up the code, execution is straightforward.
11. Would you like to have suggestions on what transformations to be used?
   I would like if the IDE shows me suggestions on what transformations to use based on the data type and source.
12. How often do you find yourself redoing the same transformation due to errors?
   a. often.
13. Do you repeat some ETL workflows often (same transformations, different data)
   a. Yes. Errors occur frequently.
14. What would you like to see changed about applying transformations on data?
   a. Show if the connections to the data source/ destination are active, show the available datasets and properties.
15. Do you receive feedback from the tool after applying transformations? (failure/success of extracting and transforming data, other errors/warnings). If yes to (15), Do you find the feedback easier to interpret?
   a. No feedback about the transformation job is available. Manually checking databases is the only way

Role: Data Scientist

1. What are the ETL tools that you have used in the past?
   a. SAS tools, Informatica, Pentaho Data Integration, Talend.
2. What tools do you find easy to work with? (briefly explain what makes them easier to use)

   a. Talend has a low learning curve and less complex user interface.

3. What tools do you find hard to work with? (briefly explain what makes them harder to use)
   a. Informatica has a cluttered UI which requires spending time navigating to find the actions required to complete an ETL job.
4. What ETL tools do you currently use?
   a. An internally developed tool which uses Apache Spark as the ETL engine.
5. What data sources do you connect to most?
   a. Amazon Redshift, S3, HDFS, MYSQL
6. Are you satisfied with the performance of the tool (currently in use) when reading and writing data? (briefly explain the reason for yes/no)
   a. We cannot see the data within the tool. I will have to go into the data source and check the metadata of the file.
7. Do you use the tool to clean/ reshape the data before transformations? If yes to (7), do you find the process easier within the tool or outside of the tool? (briefly explain)
   a. Yes. It's easier to use the tool for cleaning data than using a Jupyter notebook, But I have to write the datasets into temporary tables or files in order to make sure that the cleaning process was successful.
8. Do you find the process of applying transformations to data easier? (briefly explain)
   a. Yes. Applying transformations is easier. But having to check the results after transformations require a few more steps.
9. What are the transformations that you use frequently?
   a. Dimensionality reduction, joining datasets.
10. Are those transformations can be accessed quickly and easily?
    a. Yes
11. Would you like to have suggestions on what transformations to be used?
    a. It would be helpful if I could get suggestions on what transformations to be used based on data.

12. How often do you find yourself redoing the same transformation due to errors?
    a. Frequently.
13. Do you repeat some ETL workflows often (same transformations, different data)
    a. Yes
14. What would you like to see changed about applying transformations on data?
    a. Ability to view data before and after transformations within the tool.
15. Do you receive feedback from the tool after applying transformations? (failure/success of extracting and transforming data, other errors/warnings). If yes to (15), Do you find the feedback easier to interpret?
    a. Yes. A visual feedback is given, But It will only tell if the job succeeded or failed. I will have to trace a log to find out why the job was failed.

B

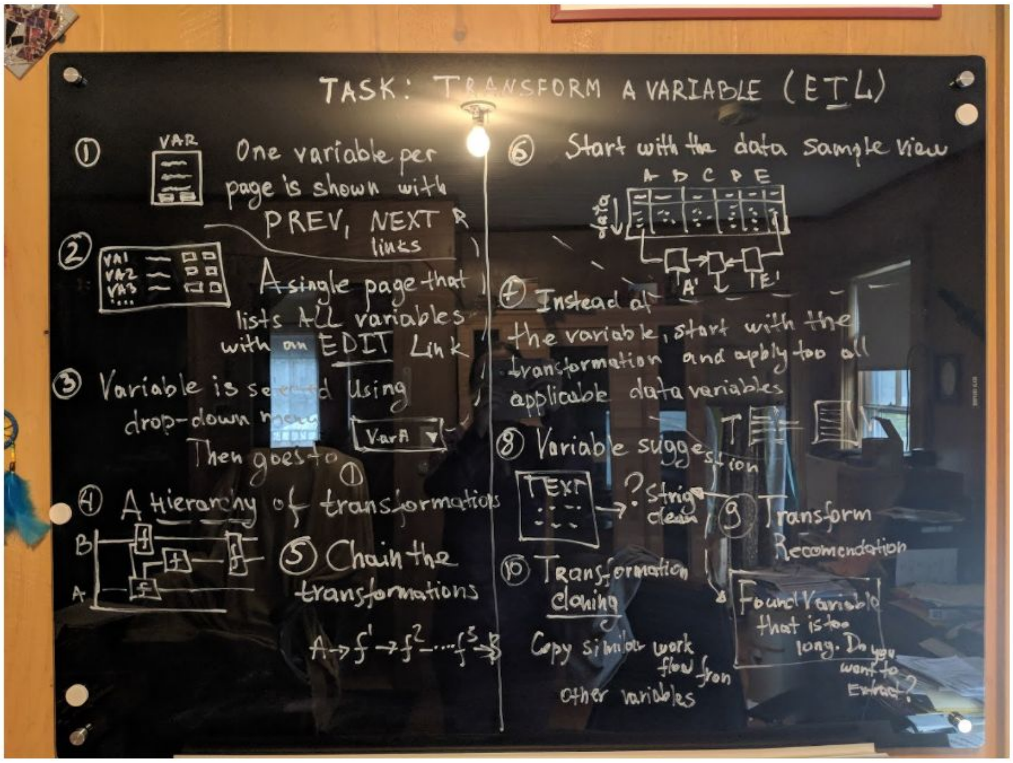
Notes From Naturalistic Observation

1. The user logs into Looker, a Business Intelligence tool.
2. The user pulls up a ticket from Jira, that says they need to create a new data visualization based on new data entering the system
3. The user spends a few minutes trying to ensure that the data has been indexed by Looker and is unable to verify that it is.
4. The user messages a co-worker who tells him that the data is in Looker, just under a different name that outlined in the ticket
5. The user opens documentation and searches for how to execute the kind of transformation he would like to perform (The user wants to create a new view that is a summarization of the raw data. They would like to take daily coming in on a second granularity and summarize it by the minute)

6. The user looks Lookml (Looker Markup Language) to construct a query that would summarize the data as desired
7. The user checks the result data and it is not summarizing correctly, more precisely there is less data than the user expected
8. The user consults the LookML documentation to ensure they are performing the right query and discovers they are not
9. The user tries a second time and has some issues with the “group by” conditions
10. The user consults LookML again and figures out the mistake they made
11. The user corrects the mistake and investigates the resulting data to their satisfaction

C

Prototype Brainstorming



TASK: TRANSFORM A VARIABLE (ETL)

① One variable per page is shown with PREV, NEXT links

② A single page that lists ALL variables with an EDIT Link

③ Variable is selected Using drop-down menu Then goes to ①

④ A Hierarchy of transformations

⑤ Chain the transformations

$A \to f^1 \to f^2 \cdots f^3 \to B$

⑥ Start with the data Sample view

⑦ Instead of the variable, start with the transformation and apbly too all applicable data variables

⑧ Variable suggestion

? String clean

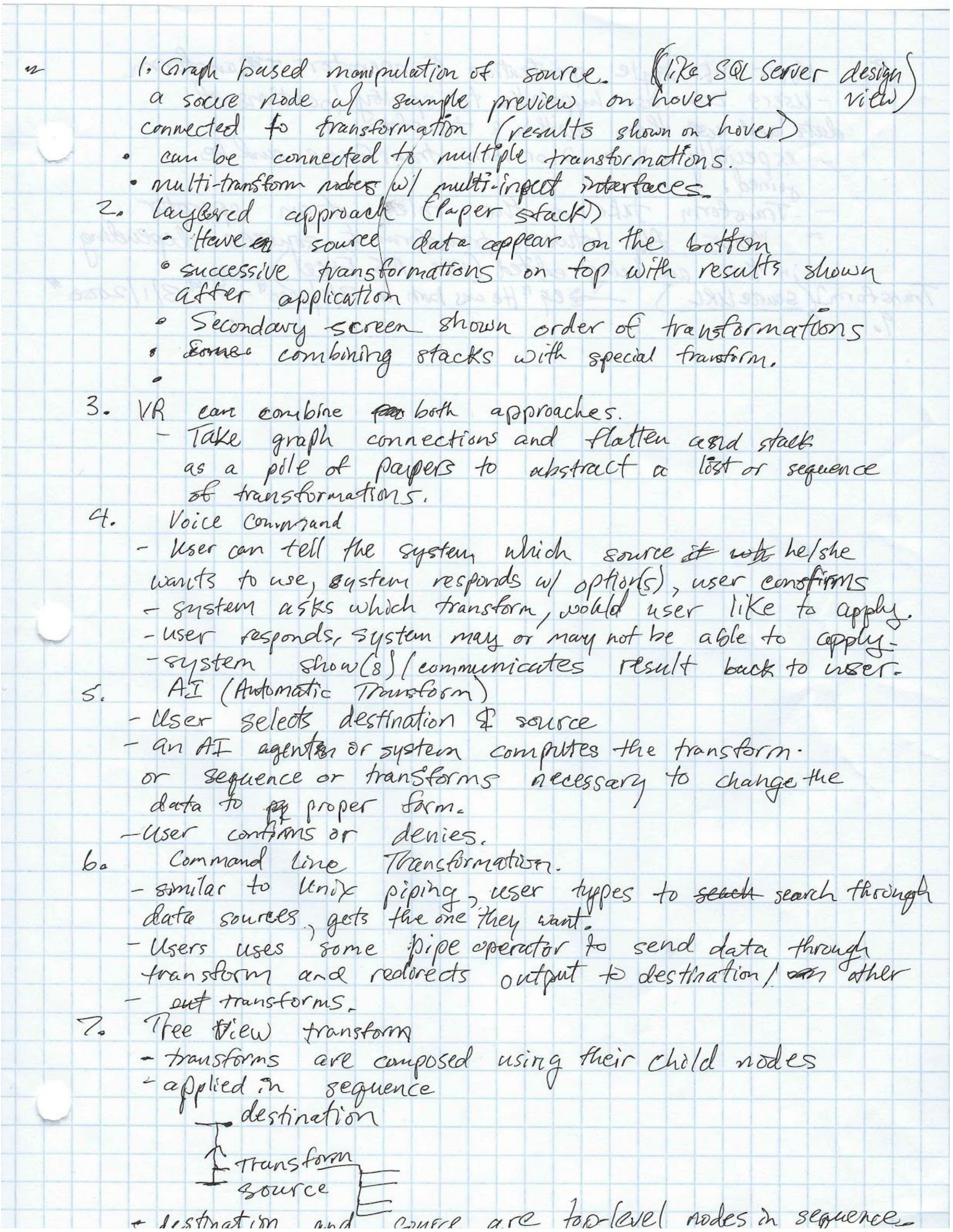
⑨ Transform Recomendation

Found Variable that is too long. Do you want to Extract?

⑩ Transformation cloning

Copy similar work flow from other variables

Figure 1: Results of the individual brainstorming session for the task of a variable transformation.



1. Graph based manipulation of source. (like SQL Server design view)
a socre node w/ sample preview on hover
connected to transformation (results shown on hover)
- can be connected to multiple transformations.
- multi-transform nodes w/ multi-input interfaces.

2. Laybered approach (Paper stack)
- Have ~~an~~ source data appear on the bottom
- successive transformations on top with results shown after application
- Secondary screen shown order of transformations
- ~~Some~~ combining stacks with special transform.
-

3. VR can combine ~~for~~ both approaches.
- Take graph connections and flatten asnd stack as a pile of papers to abstract a list or sequence of transformations.

4. Voice Command
- User can tell the system which source ~~it wh~~ he/she wants to use, system responds w/ option(s), user confirms
- system asks which transform, would user like to apply.
- user responds, system may or may not be able to apply.
- system show(s) / communicates result back to user.

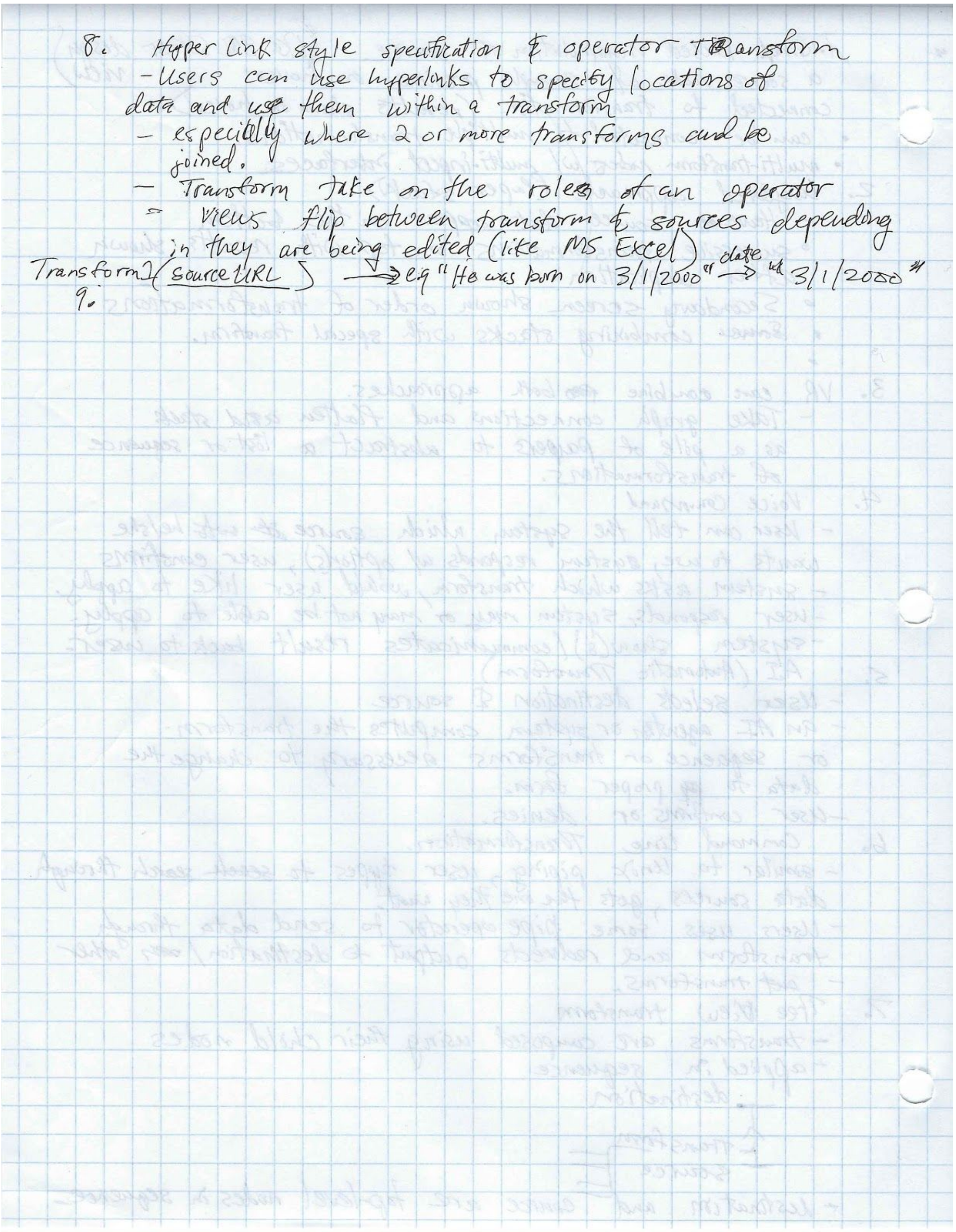
5. AI (Automatic Transform)
- User selects destination & source
- an AI agent or system computes the transform. or sequence or transforms necessary to change the data to ~~pr~~ proper form.
- User confirms or denies.

6. Command Line Transformation.
- similar to Unix piping, user types to ~~search~~ search through data sources, gets the one they want.
- Users uses some pipe operator to send data through transform and redirects output to destination / ~~an~~ other
- ~~out~~ transforms.

7. Tree View transform
- transforms are composed using their child nodes
- applied in sequence

destination
Transform
source

- destination and source are top-level nodes in sequence.



8. Hyper Link style specification & operator TRansform

- Users can use hyperlinks to specify locations of data and use them within a transform
- especially where 2 or more transforms can be joined.
- Transform take on the roles of an operator
= views flip between transform & sources depending in they are being edited (like MS Excel)

Transform1(source URL) → e.g "He was born on 3/1/2000" $\xrightarrow{date}$ "3/1/2000"

9.

Figure 2. Results of the individual brainstorming session for the task of a variable transformation.

D: Interview Questions & Answers

Question: What does your day-to-day job look like?

Participant	Answer
1	Half of my day is spend with clients the other half is spent preparing data for clients.
2	Half of my day is spent dealing with problems in the data.
3	Probably a little over half of my time is spent creating new datasets out of existing ones
4	I'd say most of my time is spent answering client related data questions
5	Most of my time is spent writing custom code to deal with data sets

Question: What's the hardest part of doing data transformations?

Participant	Answer
1	Figuring out the right dataset to work from. There is a lot of data sprawl and sometimes it's not clear if you have the right data set
2	Loops.. I have to write a lot of loops to get the data the way I want it.
3	The data size can be a problem for me. Using Excel it gets kind of bogged down with really large datasets
4	Figuring out exactly what people want. If you make one small mistake it might

	take a whole day or work to fix.
5	Writing code. I'm not a software engineer but I have to write code to get data to clients.

Question: How would you search a dataset in this prototype?

Participant	**Answer**
1	It looks like very available table is on the left in the lists.
2	I assume I can scroll the datasets on the left and find what I need.
3	It looks like all the data can be found in the dimensions and facts section of the prototype.
4	Over here [Pointing to the left panel]?
5	Looks like everything is available on the left panel.

Question: What would you expect to happen after dragging a table icon?

Participant	**Answer**
1	Nothing in particular, I assume I need to click a button before a transformation takes place.
2	I assume nothing? Not until I press another button to start the transformation.
3	It looks like it automatically tries to create a new table.
4	It looks like it will create a new table,

	unless I'm missing something
5	My guess is transformations keep happening until you stop dragging data.

Question: How difficult was it to complete the tasks asked of you today?

Participant	**Answer**
1	Not difficult at all.
2	I got it done quicker than I would with Looker!
3	Pretty easy.
4	I think I got the hang of it pretty quickly.
5	Not difficult at all.

Question: Overall, what are your thoughts on the interface?

Participant	**Answer**
1	It's really beautiful and simple.
2	Very thoughtful, it really met most of my needs.
3	I wish something like this existed in real life.
4	Thoughtful design, a few things are missing but it's pretty good for an early iteration.
5	Besides my one complaint I really don't have anything bad to say about it.

Question: What improvements would you make to the interface?

Participant	**Answer**
1	Some more fine grained control over the visualizations would have been nice.
2	Better understanding how the data could be joined and giving the user control would be welcome.
3	None that I can really tell, this seems like a great design.
4	It feels a little too magical, would be nice to see how things worked under the hood.
5	I wish I could control more details about the data is presented at the end.