

# Methods for Investigating Mental Models For Learners of APIs

We present our novel method for discovering mental models of learners of APIs. Our methods involve a background interview and a natural programming elicitation session.

Amber Horvath, Carnegie Mellon University  
Mariann Nagy, Google  
Finn Voichick, Washington University in St. Louis  
Mary Beth Kery, Carnegie Mellon University  
Brad A. Myers, Carnegie Mellon University

## Motivation

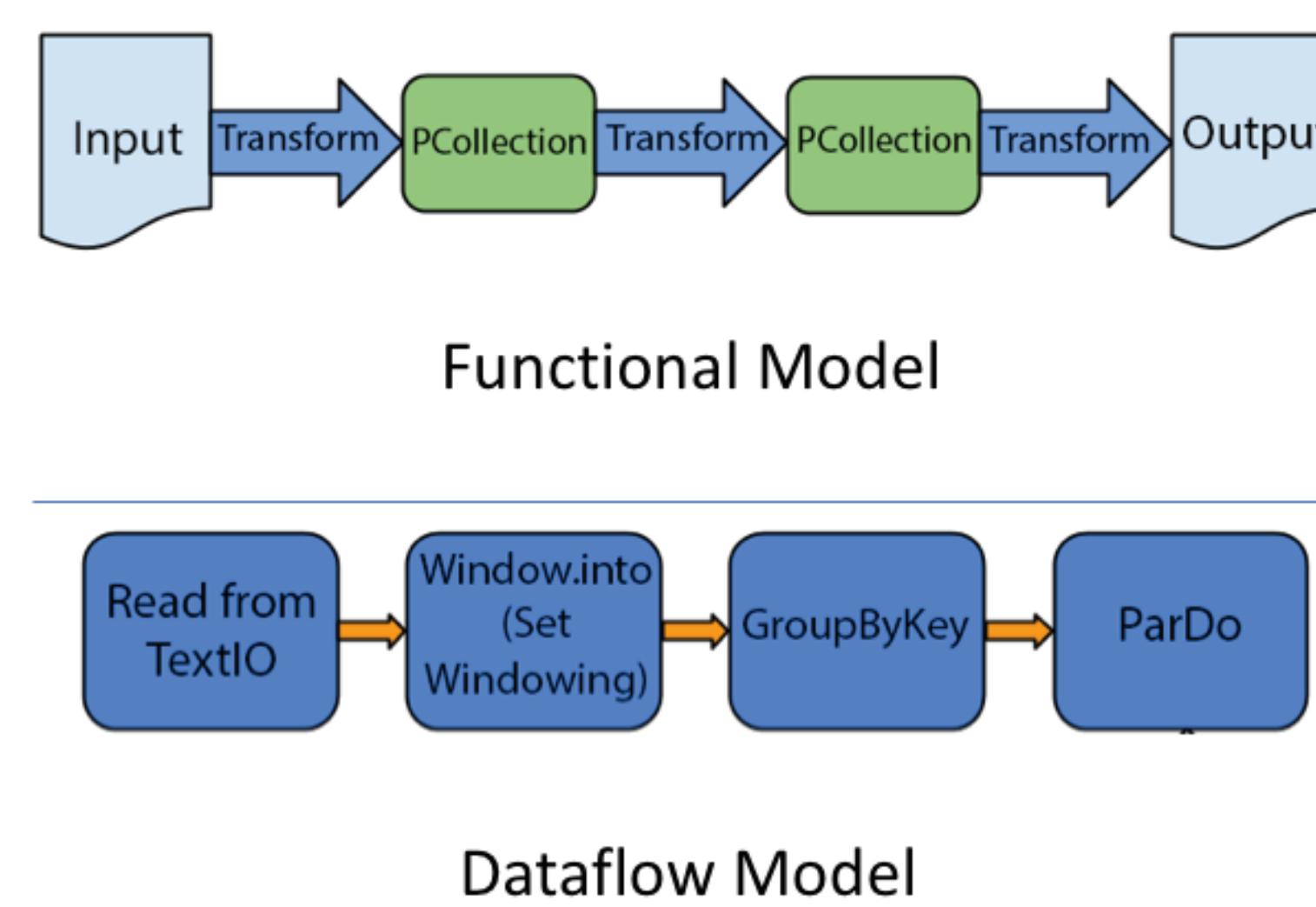
- Most software engineering requires the use of APIs.
- Little work on how to design usable APIs and how API designers may evaluate their API's usability [1].
- Some engineers end up just releasing their API as a way to test its usability [2].

Natural programming elicitation has developers express how they think about the design of a program and the constructs used by asking them to describe how they would want to call the API. We developed and evaluated our method on Apache Beam, a distributed data processing API.

- Unusable APIs have caused confusion to their users, resulting in significant security vulnerabilities [3].
- One potential way to create more usable APIs is by scaffolding a correct mental model for the user, i.e., their conceptual understanding of how a system works [4].

## The Methods

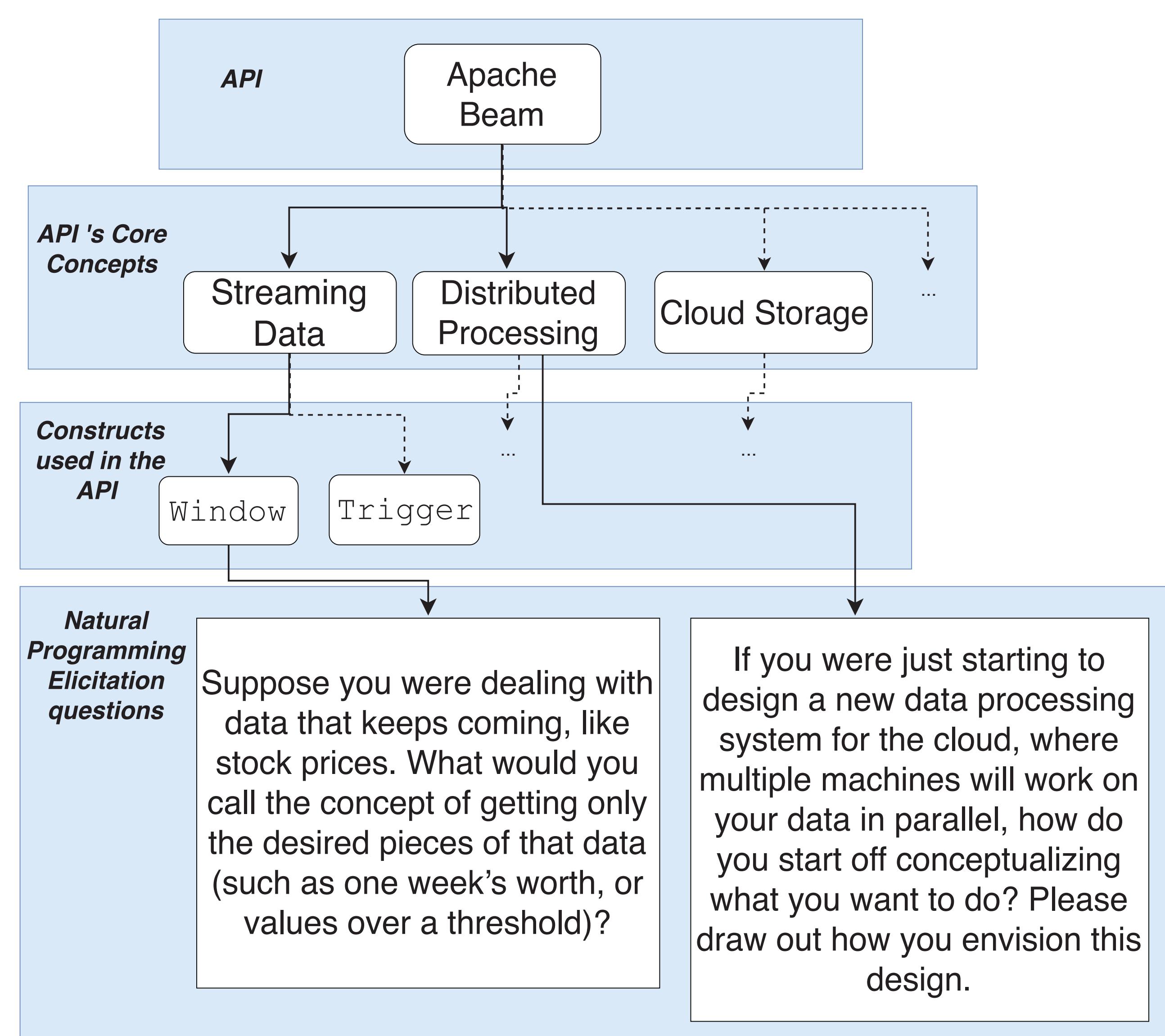
- We aim to understand what the user's mental model of the programming paradigm is through natural programming elicitation.
- We found Beam supports conflicting mental models through inconsistent diagrams (see Figure 1).
- We investigate how understanding a user's mental model can result in modifications to the API and its documentation.



**Figure 1:** Beam's documentation. The functional model has each node represent a collection of data, while the edges represent actions performed on the data. The dataflow model is the opposite with the nodes representing actions and the edges representing data.

## Using the Methods

### Generated Natural Programming Elicitation questions...



**Figure 2:** To identify mental models, we have participants describe how they would begin designing a data processing system as this results in a high level architecture representing how a participant wants to structure their program logic by answering these abstract questions.

...then compared two pilot participants' answers to the current Beam API's programming models and its chosen terminology

Pilot 1, describing how they think about their data, implying a dataflow mental model

*"I guess in terms of the applications I'm used to... we might have, say, multiple machine learning or data processing nodes in that graph..."*

Beam's name	P1	P2
PTransform	Apply	Transform, Normalize
Window	Filter	Batched, Snapshots
Combine, GroupByKey	Join, Merge	Join, Merge

**Figure 3:** Preliminary analysis of the terminology the participants used in the natural programming elicitation questions showed almost no similarity to the names used for corresponding functionality in Beam. This suggests that perhaps Beam's function names may not match what users expect.

## Current Work

Considering the difference in preferred terminology between participants, we are interested in how this mismatch affects how people find candidate API methods in Beam.

To investigate this question, we are currently expanding our work to understand how user's mental models, along with their programming styles and foraging styles, affect the discoverability of Beam's functionality.

### References

- [1] Brad A. Myers and Jeffrey Stylos. 2016. Improving API usability. *Commun. ACM* 59, 6 (2016), 62–69. <https://doi.org/10.1145/2896587>
- [2] Lauren Murphy, Mary Beth Kery, Oluwatosin Alliyu, Andrew Macvean, and Brad A. Myers. 2018. API Designers in the Field: Design Practices and Challenges for Creating Usable APIs. In *Symposium on Visual Languages and Human-Centric Computing (VLHCC '18)*. IEEE, 249–258.
- [3] Sascha Fahl, Marian Harbeck, Perl Henning, Markus Koetter, and Matthew Smith. 2013. Rethinking SSL Development in an Appified World. In *Proceedings of the 2013 ACM SIGSAC conference on Computer and Communications Security (CCS '13)*.
- [4] Don Norman. 2013. *The Design of Everyday Things*. Basic Books, New York, NY, USA, 26, 31.