

# On the Inconsistency in Grammar Evaluations

Lemara Williams 

**Abstract**—Current evaluation methods for visualization grammars are fragmented and inadequate. This work explores the community's lack of a standardized approach to evaluating grammars and comparing them. We propose broadening the evaluation criteria to include metrics like ease of use, conciseness, expressivity, and more rigorous evaluation techniques.

**Index Terms**—Grammars, Evaluation

---

## 1 INTRODUCTION

In data visualization, domain-specific languages (DSLs) or grammars have exploded in popularity. The main concept behind their use is to make some things easy by making others hard. From general-purpose grammars for web interactivity like Vega-Lite [24] to grammars for task-specific use cases like Cicero [11] for responsive visualization, the space of visualization grammars has expanded and evolved. The expansion of the field highlights the benefits of DSLs, especially around expressiveness and portability. However, their expanded use within the field has brought up the question of evaluation. With the growing number of grammars available to researchers and data analysts, how would you know which is best suited to your needs?

There is no standard process for evaluating new grammars or for grammar comparison. This is by no means a new problem; this topic has been discussed within the software engineering field for decades [7, 9, 10, 13]. Yet, with no definitive path to an answer within the originating field, how can we, in the visualization community, expect to develop evaluation metrics that serve the entire visualization field?

Data visualization, a discipline within computer science which has historically emphasized quantitative evaluation and original research, also exists within human-computer interaction (HCI), a field that places greater emphasis on qualitative evaluation. In the last two decades, only three papers proposing new JSON-based grammars have provided detailed evaluations, either through traditional DSL approaches or ad hoc reasoning [16]. To address the differing standards of these disciplines and the limited evaluation currently seen in the field, a broader shift in how visualization grammars are evaluated in research is necessary.

This paper will explore various evaluation methods for grammars and metrics, as well as techniques to enhance their effectiveness.

## 2 CURRENT EVALUATIONS

Even with traditional DSL evaluations, there is no comparable way to test the usefulness of a grammar at scale. This becomes more apparent when accounting for different audiences.

Within traditional DSL literature, evaluations are typically subjective [23] and depend on the software engineer's understanding of the grammar they developed and its usefulness in different situations.

Usability heuristics in HCI depend on a developer's ability to objectively analyze the failures and strengths of the grammar they have created. They also depend on an expert's ability to understand that information and apply it for their own project's benefit. An extra burden is placed on potential users of a grammar to consider the various complexities of these notional systems before even using them [14].

The most common of these heuristics is the Cognitive Dimensions of Notations framework [3], which supports the iterative nature of the design process by providing "discussion tools" to shape the conversation around analyzing designs. At the time of publication, Blackwell

et al. discussed 14 dimensions that developers should choose from for an activity the system is designed to support. It has been widely used within the HCI community and in software development, providing a shared vocabulary, a pre-prototyping analysis methodology, and applicability across various media and domains. As the dimensions to be analyzed are at the author's discretion, it is worth noting the common dimensions used for the analysis of visualization grammar tools.

Papers that have used the cognitive dimensions have a common analysis of viscosity (resistance to change), visibility (ability to view components easily), closeness of mapping (closeness of representation to domain), hidden dependencies (important links between entities are not visible), and hard mental operations (high demand on cognitive resources). Besides the common dimensions, their full analyses end the same way; the proposed grammar performs favorably across the dimensions, but future user studies are needed for further assessment. Blackwell et al. intended this framework to help developers understand the trade-offs in design decisions and how to mitigate them by comparing different dimensions. Existing evaluations frequently do not describe the trade-offs they have identified in their systems using this framework. Nor, do they summarize their work without fully outlining competing factors.

Alongside authors who do not utilize the framework to its full potential, its use in the field has exemplified some common criticisms of the framework. In a follow-up to the original publication, Green et al. [8] acknowledge the vagueness and overlap of some of the dimensions. One being that role-expressiveness (the purpose of an entity is readily inferred) and Closeness of Mapping are often "readily confused."

Other dimensions, such as viscosity, can be interpreted from various perspectives, leading to inconsistent or misinformed analyses. The original definition conflated two types of viscosity [8]: cognitive (resistance to a user's expectation) and structural (resistance to underlying information structure).

Nebula [5] approaches viscosity from a structural perspective, emphasizing underlying information structure rather than user actions. It is argued that the enforced template structure shortens edit distance, but this is only possible because of the many coordinated changes occurring behind the user interface. Vega [25] takes the same approach. The focus of its discussion of viscosity is on couplings and abstractions, which do make the editing process easier, but are still structural in nature. This is in contrast to the Probabilistic Grammar of Graphics [20], which emphasizes the number of user actions and independent editing of components. The focus is on the user editing process, which aligns with cognitive viscosity.

Although the Cognitive Dimensions of Notations framework is also intended as a comparison tool, researchers often compare their grammar to a similar one for various reasons, such as motivations, hidden dependencies, or other task-specific values. For instance, AnnoGram [21], an annotation grammar extension to Vega-Lite, compares its grammar against eight different tools. Some of these tools are specifically designed with annotation support in mind, while others are general-purpose languages. The authors conducted a comparative evaluation based on several metrics, including total lines of code, first-class support

---

• Lemara Williams is with Washington University in St. Louis. E-mail: lemaraw@wustl.edu

for annotations, intuitiveness, error proneness, and portability. While some of these metrics are based on prior work, others are subjectively evaluated by the authors. Error proneness, a dimension captured in the Cognitive Dimensions of Notations [3], has some background in its consistency as a metric in cognitive reasoning. However, other metrics lack justification for their relevance to potential users of the grammar. This lack of user consideration is also evident in how developers use examples as an evaluation technique.

Another common method of evaluation is the use and re-creation of examples. Designers often rely on examples for ideation and idea transfer [2]. There is merit in demonstrating how a proposed grammar can more effectively represent chart designs that are familiar to designers. Example-based evaluations can quickly showcase a grammar's unique features. Gemini [12] takes a formal approach to this by conducting a user study to replicate animated transitions authored in D3 [4]. The results provide evidence of the grammar's strengths and limitations. Other proposed grammars, like GoFish [19], walk through examples while comparing the code to how the same example would look in another language or system. GoFish also provides an example gallery of different graphics to showcase its expressiveness.

Showcasing the limitations of a proposed grammar through examples can assist designers in assessing whether it aligns with their design requirements. When a designer encounters an example that resonates, it can serve as both inspiration and a practical tool for conveying the intended task or data type. This example can act as a template for future designs. However, it is important to clarify the difference between evaluation and demonstrations. A demonstration is an act of showing that something works.

In contrast, an evaluation is a systematic process to determine merit, worth, value, or significance [1]. Using examples demonstrates a grammar. Demonstrations of what a grammar can do are not, in themselves, an evaluation method. It lacks the rigor needed to assess the significance of a grammar. That does not necessarily answer how it can serve practitioners or how it is better than what practitioners did before the grammar existed.

### 3 METRICS OF VALUE

It is worth taking the time to consider the metrics of value for different populations based on their priorities. The metrics that researchers weigh in their choice of grammar may differ from those that practitioners value more. This section considers three metrics that can enhance the analysis of a grammar and deserve greater consideration: ease of use, conciseness, and expressivity.

#### 3.1 Ease of Use

Grammars are designed to simplify user interaction, either by reducing abstraction levels or enabling faster prototyping. Prior work, such as Vega and Vega-Lite, has emphasized this as a key aspect of proposed grammars. However, instead of merely promoting reduced abstraction, the focus should shift to the user. Authors of proposed grammars often overlook the question of their target audience. While there is undoubtedly appeal to the broader research community, there is an implicit assumption that practitioners will readily adopt these tools to create visualizations for the general public.

Two core aspects to consider when discussing ease of use for a target audience are extendability and adoptability. If a grammar is proposed to fill a gap or address a specific requirement, developers should frame it in a way that illustrates its practical application within a real-world workflow. Extendability, a metric that assesses the user experience of utilizing a proposed grammar, should consider the current workflow of a potential user. Instead of overhauling it entirely, grammars should aim to accommodate the user's existing process. Consequently, extendability would be evaluated based on its ability to seamlessly integrate into existing tools. Adoptability is closely related to extendability. It assesses the learning curve for consistently using a proposed grammar. Some grammars and tools, like D3, have a high barrier of entry, requiring general data-binding knowledge, JavaScript, and web development knowledge. On the other hand, tools like Vega-Lite have a lower barrier of entry, with the code revolving around JSON specifications.

These two factors determine not only the impact of adopting a new grammar but also the level of commitment required to create proficient charts while using that grammar. They directly explain how easy it would be for a new user to utilize a proposed grammar.

#### 3.2 Conciseness

There is a delicate balance between developing grammars that are concise yet expressive. Conciseness is rooted in modularity. Simple components work together to build complex charts. This process should be easy and allow for flexible customization. Composability is one method to achieve this goal. Independent units that enable composability give users the tools to build complex charts without getting bogged down in the finer details. It also allows users to create various charts without spending too much time learning the technical details.

A simple way to examine composability within a grammar is to take two arbitrary components and see whether they can combine directly or require a specialized set of steps to achieve that combination. Vega-Lite demonstrates this well, as conciseness is one of the tenets of the grammar. The composite views created through layering, faceting, and concatenation easily support complex charts and views. However, its predecessor, Vega, requires manual reworking for shared scales to achieve a similar composite view. These two grammars exemplify the balance between conciseness and expressiveness. Vega-Lite sacrifices expressiveness for conciseness while Vega does the opposite.

Conciseness enables simple, flexible customization. A thorough evaluation of this metric helps determine the effort required to create, modify, and maintain visualizations.

#### 3.3 Expressivity

While conciseness values modularity, expressivity values robustness. When discussing expressivity, it is crucial to differentiate between the interpretations of the concept as presented by two distinct schools of thought. Formal DSL theory posits that a class possesses numerous methods, and the resulting graph repertoire is a product of the entire set of these methods [6]. As such, a collection of objects can generate a wide range of intricate and diverse chart types. Key concepts emerging from this school of thought include expressive power, which refers to the ease with which desired outcomes can be translated into practice. Elegance, which pertains to the creation of human-readable code. And extensibility, which involves the ability to effortlessly extend code to accommodate additional aspects of a task or language. Pu and Kay [20] employ this evaluation metric to discuss the limitations of their grammar.

On the other hand, perceptual theory within visualization, as detailed by Munzner [17] and Mackinlay [15], focuses on expressivity in encoding. The main question is whether the encoding is semantically valid for the data type being used. This also carries the idea that correctness is not equivalent to effectiveness. However, this is not equally enforceable on all grammars. Ones that do not have a concept of a data-type model like D3 place the full burden of correctness on the user. Conversely, grammars with enforceable constraints on data types, such as Vega-Lite, have correctness built in. A grammar like D3, while being expressive under formal DSL theory, would be considered less expressive under perceptual theory because of the possibility of creating "incorrect" charts.

Grammars can score well on correctness while producing ineffective charts. This conflicting notion of expressivity makes it challenging to compare grammars. To fully capture a grammar's expressiveness, it would be prudent to consider both aspects of expressivity in the future.

### 4 FUTURE TECHNIQUES

As we continue to develop visualization grammars and evaluation techniques, we can implement methods as a community to create consistency in evaluating grammars.

One such method is creating a benchmark repository. A similar idea has been suggested for visualization authoring systems by Ren et al. [22] and implemented in the Visual Analytics Benchmarks Repository for the InfoVis contest [18]. A metric comparison system proposed

by Kruchten et al. [14] based on examples would also fit this description. The idea behind a benchmark repository is that it can serve as a centralized hub for the wider research community to see how others have evaluated their grammar and aid designers in finding metrics most relevant to determining which grammar is most applicable. Metrics proposed earlier would also fit within the repository. The largest contribution is that having a central hub would aid in defining metrics that have reached consensus on how they would be evaluated.

Another method to aid in consistency is user testing and expert evaluations. This is not a new technique in HCI, nor is it being applied to the evaluation of visualization grammars. However, they are rarely utilized as an evaluation tool in the subfield. There are valid reasons for that. Grammars are inherently predisposed for use by those more experienced within a field. It may be difficult to recruit users with the background needed to engage in meaningful interactions with a proposed grammar. It may also be difficult to develop a task that can yield significant results. However, defining the audience for a grammar and getting feedback from that audience is important for understanding not only how people will use it but also how it should be tested. Regarding tasks, even cognitive walkthroughs for specific task flows can reveal strengths and limitations of a grammar that the developers may have overlooked. Feedback from outside of the development cycle is invaluable.

As the field of visualization grammars progresses, consistent and rigorous evaluation techniques will be crucial for meaningful comparisons between grammar designs.

## 5 CONCLUSION

This work delves into the fragmented and inconsistent nature of visualization grammar evaluations. While common evaluation techniques like the Cognitive Dimensions of Notations framework, comparative analysis, and re-creation of examples excel in qualitative analysis, they often lack the rigor required to assess the significance of a grammar or do so inconsistently. As the field of visualization continues to evolve, it is crucial to consider metrics and techniques that prioritize user-centric approaches and promote collaboration. If evaluation of grammars continues to rely on inconsistent metrics that fail to consider the user, grammars will fail to be meaningfully compared, which will limit progress within the field.

## REFERENCES

- [1] American Evaluation Association. What is evaluation? <https://www.eval.org/Portals/0/What%20is%20evaluation%20Document.pdf>. 2
- [2] H. K. Bako, X. Liu, G. Ko, H. Song, L. Battle, and Z. Liu. Unveiling how examples shape visualization design outcomes. *IEEE Transactions on Visualization & Computer Graphics*, 31(01):1137–1147, 2025. 2
- [3] A. Blackwell and T. Green. Notational systems—the cognitive dimensions of notations framework. *HCI models, theories, and frameworks: toward an interdisciplinary science*. Morgan Kaufmann, 234, 2003. 1, 2
- [4] M. Bostock, V. Ogievetsky, and J. Heer. D<sup>3</sup> data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011. 2
- [5] R. Chen, X. Shu, J. Chen, D. Weng, J. Tang, S. Fu et al. Nebula: A coordinating grammar of graphics. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4127–4140, 2021. 1
- [6] M. Fowler and R. Parsons. Domain-specific languages, 2010. 2
- [7] P. Gabriel, M. Goulão, and V. Amaral. Do software languages engineers evaluate their languages? pp. 149–162, 2011. doi: 10.48550/arxiv.1109.6794 1
- [8] T. R. Green, A. E. Blandford, L. Church, C. R. Roast, and S. Clarke. Cognitive dimensions: Achievements, new directions, and open questions. *Journal of Visual Languages & Computing*, 17(4):328–365, 2006. 1
- [9] B. Hoffmann, N. Urquhart, K. Chalmers, and M. Guckert. An empirical evaluation of a novel domain-specific language – modelling vehicle routing problems with athos. *Empirical Software Engineering*, 27, 2022. doi: 10.1007/s10664-022-10210-w 1
- [10] S. Joel, J. Wu, and F. H. Fard. A survey on llm-based code generation for low-resource and domain-specific programming languages. *ACM Transactions on Software Engineering and Methodology*, 2024. doi: 10.1145/3770084 1
- [11] H. Kim, R. Rossi, F. Du, E. Koh, S. Guo, J. Hullman et al. Cicero: A declarative grammar for responsive visualization. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2022. 1
- [12] Y. Kim and J. Heer. Gemini: A grammar and recommender system for animated transitions in statistical graphics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):485–494, 2020. 2
- [13] P. Klint, R. Lämmel, and C. Verhoef. Toward an engineering discipline for grammarware. *ACM Trans. Softw. Eng. Methodol.*, 14:331–380, 2005. doi: 10.1145/1072997.1073000 1
- [14] N. Kruchten, A. M. McNutt, and M. J. McGuffin. Metrics-based evaluation and comparison of visualization notations. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):425–435, 2023. 1, 3
- [15] J. Mackinlay. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141, 1986. 2
- [16] A. M. McNutt. No grammar to rule them all: A survey of json-style dsls for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):160–170, 2022. 1
- [17] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2014. 2
- [18] C. Plaisant. Visual analytics benchmark repository, 2006. 2
- [19] J. Pollock and A. Satyanarayan. Gofish: A grammar of more graphics! *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2
- [20] X. Pu and M. Kay. A probabilistic grammar of graphics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020. 1, 2
- [21] M. D. Rahman, M. Rahat-Uz-Zaman, A. McNutt, and P. Rosen. Annogram: An annotative grammar of graphics extension. In *2025 IEEE Visualization and Visual Analytics (VIS)*, pp. 236–240. IEEE, 2025. 1
- [22] D. Ren, B. Lee, M. Brehmer, and N. H. Riche. Reflecting on the evaluation of visualization authoring systems: Position paper. in 2018 IEEE evaluation and beyond-methodological approaches for visualization (beliv). 86–92, 2018. 2
- [23] I. P. Rodrigues, M. B. Campos, and A. Zorzo. Usability evaluation of domain-specific languages: A systematic literature review. pp. 522–534, 2017. doi: 10.1007/978-3-319-58071-5\_39 1
- [24] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016. 1
- [25] A. Satyanarayan, K. Wongsuphasawat, and J. Heer. Declarative interaction design for data visualization. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pp. 669–678, 2014. 1