Applying the Cognitive Dimensions of API Usability to Improve API Documentation Planning

Robert Watson
Department of Human-Centered Design & Engineering
University of Washington
Campus Box 352315, Seattle, WA 98195
01-206-685-1557
rbwatson@uw.edu

ABSTRACT
This interactive poster explores the application of the 12 cognitive dimensions of API usability to API documentation planning by using the dimensions to identify and characterize the factors that influence the documentation that the users of an API require. Many factors can complicate estimating and planning the documentation an API requires. Even when an API's documentation requirements can be estimated, it can be difficult to present to stakeholders an objective basis for the estimate. The cognitive dimensions of API usability have characterized APIs and their users successfully and they have been used to communicate these characterizations to stakeholders. It follows that the same dimensions could also help identify the documentation that an API requires to provide a satisfactory and successful experience for the software developers who use the API.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Training, help, and documentation

General Terms
Documentation, Human Factors, Theory.

Keywords
API, API reference documentation, Application programming interface, Software documentation, Software libraries

1. INTRODUCTION

The usability ideal of a product being so intuitive that it does not require any documentation applies to application programming interfaces (APIs) as much as it does to any other product. In practice, however, this goal can be difficult to attain. When an API does not live up to that goal, documentation is often used to help fill gaps that exist between an API's design and the software developer's understanding of how to use the API successfully. If it were possible to characterize those gaps objectively, it would be easier to understand, estimate, and describe the documentation required to bridge them.

SIGDOC '14, Sep 27-28 2014, Colorado Springs, CO, USA
ACM 978-1-4503-3183-8/14/09
http://dx.doi.org/10.1145/2666216.2666239

Clarke [1] used the 12 cognitive dimensions of API usability to describe usability-related characteristics of APIs and their users. He described usability issues with APIs by relating the observed differences between the cognitive dimensions of the API's design and its affordances to those of the API's target users [2]. Most of the literature about using the cognitive dimensions focuses on how to use them to inform and refine the API's design; however, when the API cannot change, that same information could inform the API's documentation requirements. Through this interactive poster, I want to begin a conversation about using the cognitive dimensions of API usability to identify the documentation requirements of an API.

2. DISCUSSION

Clarke [1] used these 12 cognitive dimensions of API usability at Microsoft to improve the usability of the .NET Framework by identifying differences between the API's design and the target user persona(s).

1. Abstraction Level
2. Learning Style
3. Working Framework
4. Work-Step Unit
5. Progressive Evaluation
6. Premature Commitment
7. Penetrability
8. API Elaboration
9. API Viscosity
10. Consistency
11. Role Expressiveness
12. Domain Correspondence

Clarke applied the cognitive dimensions to inform the API designers of usability issues he found in usability tests so the designers could then improve the API. Microsoft experienced some success using the cognitive dimensions to improve API usability. Bore and Bore [3], however, observed that, "In practice there are too many cognitive dimensions, their interpretation is often not obvious, measuring them is too time consuming and assigning quantitative measures involves subjective judgment." and set forth a simpler set of three dimensions with which to profile APIs. This review seeks a middle ground that is detailed enough to be useful and simple enough to be used. The next sections describe the different aspects involved in using the cognitive dimensions for API documentation estimation: the information that software developers expect API documentation to contain and how the cognitive dimensions influence API documentation requirements.

2.1 API Documentation

Watson, Stamnes, Jeannot-Schroeder, and Spyridakis [4] list the following elements that software developers have said they expect to find in API documentation.

- Overview documentation
- Short code snippets that use an API in context
- Code examples that show best-practices using an API

- Scenario and task-based documentation
- Meaningful descriptions [of the API elements]

For the users of API documentation, this list describes the information they seek from API documentation. For the writers of API documentation, this list describes the types of information they would provide in API documentation.

2.2 Managing the Cognitive Dimensions

To improve the usability of the 12 cognitive dimensions, I grouped them by their influence on API documentation requirements based on 10 years of writing API documentation. Table 1 lists the cognitive dimensions whose influence on API documentation varies with the API's design but is constant for all user profiles. The dimensions in this group are sub-divided into those that apply to understanding the API and those that apply to using the API.

Table 1. Cognitive dimensions whose documentation impact is based on the API design

Understanding related	Usage related
Consistency	Working Framework
Domain Correspondence	API Elaboration
Role Expressiveness	API Viscosity

An example of how a dimension might describe the documentation required is *consistency*. An inconsistent API requires more documentation than a consistent one, regardless of the target user profile's preference for consistency, because it must frequently describe and explain the inconsistencies. Likewise, the documentation required to address the API Elaboration dimension of an API (how much the API must be modified or extended before it can be used) varies with the API because different levels of API Elaboration will require different levels of detail, but the user profile's preference for API Elaboration does not change the amount of API documentation required.

The next group lists the dimensions whose documentation impact is a function of the interaction between the API and user profile. Table 2 sub-divides the dimensions in this group into those that apply to learning an API and those that relate to applying the API to a task.

Table 2. Cognitive dimensions whose documentation impact depends on the API design and the user profile

Learning related	Application related
Abstraction level	Progressive Evaluation
Learning style	Work-Step Unit
Penetrability	Premature Commitment

As an example, the documentation required for an API that is not easily penetrated, such as a closed-source API, would depend on whether the user needs to see inside (penetrate) the API in order to accomplish the task. An API that supports a functional level of Progressive Evaluation (where a task step can be tested only when a complete function has been developed) will need more documentation to describe the necessary steps for a user profile that prefers a local Progressive Evaluation (where tasks can be tested incrementally) than it would for a user profile that can work with a functional Progressive Evaluation.

3. APPLICATION

To produce the best combination of API documentation topics and information, the writer must understand the user and the tasks they will want to perform with the API in addition to the API and its design properties. The cognitive dimensions provide a framework with which to characterize the user and the API and the resulting characterizations can then be used to identify the required mix of information to provide in the documentation. Unfortunately, the matrix of cognitive dimensions that describe each user profile and API element can become very intimidating very quickly. This section proposes some possible methods to organize and manage them to make this analysis more practical.

Evaluating 12 dimensions of every element of an API would be very labor intensive, but is most likely unnecessary. A stratified sampling technique could group APIs by programming pattern or other relevant property. For example, divide the API by the programming patterns used and select representative elements (classes, methods, or functions) from each group to evaluate, and then generalize the results to the group.

A quantitative approach might attempt to tabulate the different dimensions of each user profile and API pattern and then review the differences. This method provides a framework within which to organize and analyze the data, which could improve future analysis and estimates over time. A qualitative approach might use the dimensions to provide the objective vocabulary to understand and explain the usability issues that inform the documentation requirements. Most likely, the best method will be a hybrid approach that can be applied quickly to provide data for estimating documentation requirements, providing usability information, and facilitating discussions with stakeholders.

4. NEXT STEPS

Identifying the potential of the cognitive dimensions is the first step towards developing a new method for the design and development of API documentation. Initially, I see this method helping technical writers and software developers identify and prioritize the documentation that the users of an API require to be most effective. As the method matures, it could inform API design decisions by clarifying the downstream costs of design decisions. Based on the feedback from the interactive poster session, I will continue researching this by:

1. Identifying research partners with whom to collaborate.
2. Validating or refining the assumptions of this approach.
3. Identifying the influence of each dimension on API documentation.
4. Testing to provide empirical data on the method in practice.

5. REFERENCES

[1] S. Clarke, "Describing and measuring API usability with the cognitive dimensions," presented at the *Cognitive Dimensions of Notations 10th Anniversary Workshop*, 2005.

[2] 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*, vol. 17, no. 4, pp. 328–365, 2006.

[3] C. Bore and S. Bore, "Profiling software API usability for consumer electronics," presented at the Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on, 2005, pp. 155–156.

[4] R. Watson, M. Mark Stamnes, J. Jeannot-Schroeder, and J. H. Spyridakis, "API documentation and software community values: a survey of open-source API documentation," in the Proceedings of the 31st ACM international conference on Design of communication, 2013, pp. 165–174.