

Describing and Measuring API Usability with the Cognitive Dimensions

Steven Clarke
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052
USA
stevencl@microsoft.com

1. Cognitive Dimensions to the rescue!

The Visual Studio user experience group at Microsoft is responsible for helping to design and build the user experience for developers using Visual Studio, .Net and the WinFX Application Programming Interfaces (APIs).

The relevance and utility of the Cognitive Dimensions framework [1] was proven early on in the development of the first release of the .Net APIs. We ran a usability study to determine if users would be able to use .Net to accomplish a set of tasks. The results of the study indicated that many users would face significant difficulties when working on these tasks. Clearly there was some work that needed to be done to improve the user experience around the APIs but it wasn't immediately clear exactly what that work was.

Many of the observations that were made during the usability study were of participants struggling with the API documentation. Participants spent a lot of time looking for classes that would show them how to accomplish the task. While searching through the documentation many participants stumbled upon the documentation for the classes that they ought to use. However, even after stumbling upon these classes, all participants continued to search for something else.

The immediate reaction of the team responsible for implementing the API was that the documentation needed to be fixed. The necessary classes needed to be sufficiently documented such that the connection with the task would be made clearer to users. However, the user experience team believed that the issues participants faced in the lab were deeper than simply being unable to find the appropriate documentation. In an attempt to get at the underlying cause of the problem, we used the Cognitive Dimensions framework to describe each usability issue in terms of specific dimensions. Doing so made it clear to us that

the reason that participants continued to search for other classes in the documentation was because the abstraction level of the classes that they stumbled upon in the documentation was too low. Participants expected to find classes that directly corresponded to the way that participants thought about the task. Instead, the classes they found were of a sufficiently low level of abstraction that participants could not relate these abstractions to the task they were working on.

We used the Cognitive Dimensions framework to present our analysis of the issues. By doing so, we were able to convince the development team to create classes at a higher level of abstraction that represented the task in the way that participants thought of it. Six months later we ran a follow up study with the new classes and observed a significant improvement in the user experience. Participants in this study were able to accomplish the same task in twenty minutes that participants in the previous study had been unable to complete in two hours.

Our initial success in using the Cognitive Dimensions framework motivated us to continue using it throughout the design and implementation of the WinFX APIs.

2. Using the Cognitive Dimensions

Our approach has been to modify the Cognitive Dimensions framework to enable teams at Microsoft to use the framework to describe and measure API usability. After careful review of the original framework we believed that the framework needed some minor modifications to make it more relevant to API usability and to increase the likelihood of the framework being used effectively at Microsoft.

For example, one of the changes that we made was to the 'Abstraction Gradient' dimension. In the original framework, this dimension describes the facilities available to the programmer to create new abstractions and the extent to which the programmer is forced to

create new abstractions in order to be able to complete some programming task. When discussing API usability, the facilities available to create new abstractions are orthogonal to the API. However, what is important is whether the abstractions are primitive components, aggregate components or factored components (somewhere in between primitive and aggregates). We describe this as the 'Abstraction Level' exposed by the API.

Our version of the Cognitive Dimensions is as follows:

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**

Through a variety of efforts we have encouraged teams around Microsoft to use the Cognitive Dimensions framework when designing APIs.

2.1 Introducing the Cognitive Dimensions

We regularly provide different teams with an introduction to the Cognitive Dimensions framework through a 90 minute presentation. We use highlights from usability studies to show the impact that each dimension has on the usability of a particular API. Demonstrating each of the dimensions with concrete examples from various APIs helps to make sense of the concepts represented by each of the dimensions.

During the presentation, many people recognize the particular issues that are described as ones that they themselves have experienced. It becomes clear that the framework is useful for describing common issues that arise with APIs.

Having introduced the concept of API usability and the Cognitive Dimensions framework, the next step that teams typically want to take is to use the framework to evaluate their own API.

2.2 Evaluating an API with the Cognitive Dimensions

We run usability studies on the various APIs that comprise WinFX. Participants are recruited and asked to use the WinFX APIs to write code that accomplishes various tasks. We collect various pieces of data during each study, such as video records of participant behavior and participants own verbal accounts of their actions.

The results of the studies are described in terms of the Cognitive Dimensions framework. We examine all of the issues that arise in a study and describe them in terms of the appropriate dimensions. In many cases, a single issue can be described from the perspective of more than one dimension. The advantage of this approach is that it encourages teams to consider multiple solutions for resolving problems.

2.3 Profiling users with the Cognitive Dimensions

In addition to using the framework to describe the usability of an API we also use the framework to describe the characteristics of the different users who will be using the API. We use three different user personas to represent the three main work styles that developers follow when using Visual Studio. Each of the personas represents different work styles and user characteristics. For each of the three personas we have created a profile that describes the ideal 'measurement' of each dimension for that persona.

Profiling the personas in this way provides teams with a set of usability goals for their API. When teams know the persona that they are designing for, they can use the persona profile to create a set of design goals and guidelines that will help them design a usable API for that particular persona. Without such goals teams are able to talk about API usability knowledgeably, but they are unable to determine what they need to do to make sure that the specific API that they are designing is as usable as it can be for specific customers and scenarios.

3. References

- [1] Green, T. R. G. & Petre, M. (1996) Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *J. Visual Languages and Computing*, 7, 131-174.