

## Understandability benchmark

### 1. Visibility of Dependencies:

#### D1:

The sources of dependencies explicitly depicted are:

- a) The user selects one of the layers (counties, rivers etc) and then the button to change the colors / or toggle labels is enabled.

The sources of dependencies in system are:

- a) The user selects one of the layers (counties, rivers etc) and then the button to change the colors / or toggle labels is enabled.
- b) The user selects the counties from the layers section, to be displayed on the map. These are automatically selected by the boundaries and regions subsection.

So the value of  $D1 = \frac{1}{2} = 0.5$

For D2, it takes one single click to select the dependencies for each control on the map. So to select  $n$  controls we would require  $O(n)$ .

### 2. Visibility of program structure

Depiction of Program structure answers questions such as “**What modules are there in this program?**” and “**How do these modules logically fit together?**”

The benchmark is related to **Role Expressiveness**. (What is this bit for?)

Applying this to the River basin project, there are various subparts (modules) of the project depicted on the homepage.

These sub sections can be considered as modules in the programming languages domain. These sub sections are typically represented by textual notations and are not role expressive. (Maybe include small images to depict the role of each sub section, is helpful). Another way is have a sub section come up, when the mouse is moved over that link.

Benchmark PS1 → Presence or absence of program structure information in S.

**PS1 is not present.**

Benchmark PS2 → Number of steps required by a programmer to navigate to this information.

**PS2 = NA.**

### 3. Visibility of Program logic.

Benchmark L1 measures whether S provides visibility of the fine-grained program logic and benchmark L2 measures the number of steps to navigate to it.

**L1 in our project is “yes”.**

Consider the Maps → Layers section. This section shows the program logic in the static view. Showing the counties, rivers, interstate is nothing but depicting fine-grained logic.

**Number of steps required is O (1).**

Similarly, clicking the Find Region sub section, the program logic is visible and the worst case number of clicks required is 2 per region. So finding details about n regions the number of clicks required are O(n)

**L3 → N/A**

### **3. Display of Results with Program Logic:**

This group of benchmarks measures **whether it is possible to see a program’s partial results displayed with the program source code.**

**Benchmark R1** measures whether or not it is possible to see the results displayed statically with the program source code

**R1 in our project is “yes”.**

**Benchmark R2** measures the number of steps required to do so.

R2: O (1)

### **Secondary Notation :- non-semantic devices.**

Benchmark SN1 measures the presence of secondary notational devices.

Benchmark SN2 measures the number of steps required to navigate to instance of them.

- a) Optional naming or labeling. (Yes)
- b) Layout of a program in ways that have no semantic impact. (yes)
- c) Textual annotations and comments. (Yes)
- d) Static graphical means of documenting a program. (No)

SN1 = 3 / 4 = 0.75

SN2: O (n)

4 steps are required to add a label to a portion of the program. So for adding “n” labels, I will require 5n steps. So the order is O (n).

The number of steps required for textual annotations is 1. So for n textual annotations we will require O (n).

## Scalability Benchmarks

### 1. Abstraction Gradient.

Benchmark AG1 measures the sources of details that can be abstracted away from a representation

Benchmark AG2 measures the number of steps required to do so. AG1 is a ratio.

For the denominator we identified 4 sources of details that might be abstracted away in a representation:

- a) Data
- b) Operations
- c) Other fine –grained portions of the program
- d) Details of navigational instrumentation.

In the River valley basin project,

The Tools (which allow us to carry out various operations on the map) have been abstracted. (Operations)

A single click is required to abstract away the tools.

The Data can also be abstracted away in the maps → Layers section.

The contents can be abstracted by clicking the folder icon. This also requires a single click.

1. Data → 1
2. Operations → 1
3. Other fine grained portions of the program → 0
4. Details of navigational instrumentation devices → 0 ( the scroll bar cannot be abstracted).

**AG1: 2/4 → 0.5**

**Benchmark AG2: number of steps required to reach AG1**

In our project AG2 is O (1)

### 2. Accessibility of related information

Any 2 pieces of information in a program are related if the programmer thinks they are.

Based on the **Visibility and Side-by-Side Ability CD**, the benchmarks in this group measure a programmer's ability to display desired items side by side.

**Benchmark RI1** measures whether it is possible to include all related information in S

**Benchmark RI2** measures the number of steps to navigate to it.

RI1 is "yes".

RI2:  $O(1)$  to display information about one city. So for  $n$  cities, the number of clicks is  $O(n)$ .

### **3. Use of Screen real estate.**

**SRE1:** Is the maximum number of program elements that can be laid out on a physical screen.

Assuming a normal screen size, we can see the toolbar, the Layers pane, the maps pane, the various tabs to go to different sections (navigation bar), from the Maps→Layers section. The drop down list in the colors section can be replaced with multiple check boxes / radio buttons, so prevent non-semantic intersection. So setting the border color, width of the line, type of the line etc can be added as a component. So the SRE1 value can be 6.

**SRE2:** is the number of non-semantic intersections that can be counted on the layout chosen in performing benchmark SRE1.

We choose, not to have any semantic intersection and then apply SRE1.

#### **Audience specific RB's**

**Whether the audience specific languages representation is well suited to its particular audience.**

The first step to carry out is identifying the audience. In our project, we identify a learned user as a target audience. This might seem as a broad classification, but if we apply to the maps section of the project, then I think the classification is not that broad.

#### **AS1: The Objects.**

In this project, the user interface primitives look like those in the audiences background. Interfaces for search, locating a place and displaying information on the map match the background pretty well.

There is hardly any use of grids and tables in maps, but where ever they have been used they match the audience background.

The active identify layer is not too obvious, and hence I feel is it not in line with the audience background.

So the value of **AS1** =  $2 / 3$

### **AS2: The operations**

The search operations are not in line with the intended user's background. Changing the background and various colors can be considered as an operation and is not so intuitive to the user.

So  $AS2 = 0 / 2 = 0$ .

### **AS3: Spatial composition.**

SQ1: Does the layout of objects and operations look like the way these objects and operations are laid out.

Yes.

SQ2: Do the lines connecting these objects are connected as in the intended audiences background?

No.

So  $AS3 : \frac{1}{2} = 0.5$