How to use Enriched Dependency Structural Matrix

Jannik Laval
RMoD Team
INRIA - Lille Nord Europe - USTL - CNRS UMR 8022
Lille, France
jannik.laval@inria.fr

Abstract—This document explain the fundamental information provided by eDSM

Note for the reader: this paper makes heavy use of colors in the figures. Please obtain and read a colored (e.g. on-line) version of this paper to better understand the ideas presented in it.

1 DSM Presentation

Dependency Structural Matrix (DSM) is an adjacency matrix: a cell represents a link between two packages. The rule for reading the matrix is: an element in column header references an element in row header at the crossing of the column and the row. For example, in Figure 1, A references B and C, B references A, C references D and D references C.

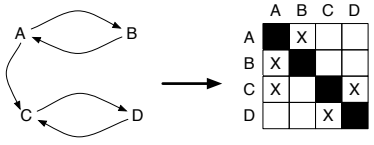

Fig. 1. A simple dependency graph and its matching DSM

In this visualization, the studied elements are packages. Previously in this tutorial, we talk about package rather than element.

2 DSM Cell color

A DSM is represented with numbers and several colors Figure 2. Numbers represents the sum of all dependencies between the two involved packages. Color is an help system to understand the structure:

- gray cells: represent a dependency between a source package and a target package
- blue squares (covering several cells): represent a **strongly connected component (SCC)**, all packages involved in a SCC are in cycle
- yellow cell: represent a single dependency involved in a SCC.
- red/pink cell: represent a cycle between two packages ($A \rightarrow B$ and $B \rightarrow A$), we name it *direct cycle*. If a cell is red, it means that the number of dependencies from column to row is a third (or less) than the number of dependencies from row to column (note: the symetric cell will be in pink). It means that this dependency is probably easier to remove than its counterpart.

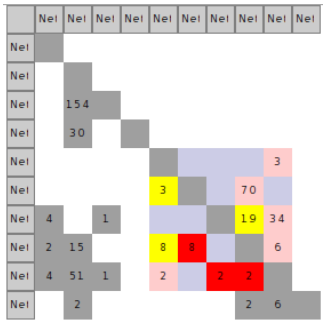

Fig. 2. Color in DSM: red/pink $\rightarrow$ direct cycle, yellow $\rightarrow$ a link composing a SCC.

3 Enriched contextual cell information

A cell can be *opened* to give more details on the dependencies. We call it an *eCell* It is presented as Figure 3. An eCell is written from top to bottom:

- on top, there is the number of dependencies. It shows the total number of dependencies, and the number or four different types of dependencies (inheritance, reference, invocation, extension).
- In the core of the eCell, we show the classes involved in the dependency and the links between these classes. A class can have three colors and one specific border. Specific colors are: pale orange means the class is involved in other dependencies, strong orange means a method of this class is involved in other dependencies, gray is for other classes. A black thick border means that the class is implied in both dependencies of a direct cycle. A dotted border represents a class

extension: the class is not defined in the targeted package
- at the bottom, there is the state of the dependency, which is the color of the cell presented in Section 2.

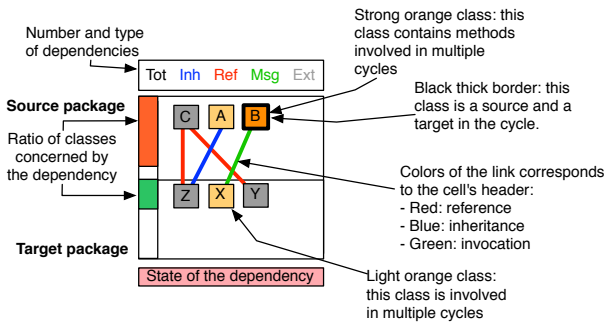

Fig. 3. Enriched cell structural information.

3.1 Tooltip

When the cursor is on a non-empty cell: a tooltip appears with an eCell inside. If it is on a direct cycle, it shows the two concerned dependencies, as in Figure 4

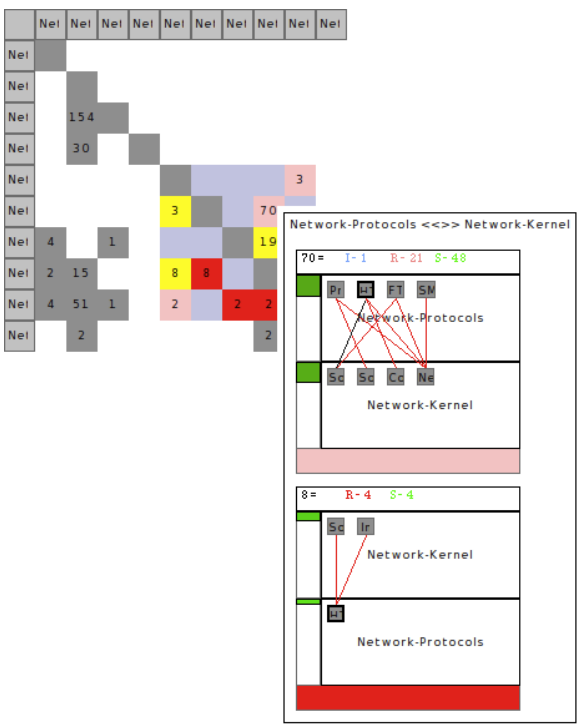

Fig. 4. Fly by help in action.

3.2 Contextual menu and double click

By right clicking on colored cells (blue, red or yellow), it is possible to open a DSM or a DSM with eCell focusing the specific SCC. In Figure 5, we can see the contextual menu, which provide, for example a DSM as in Figure 6.

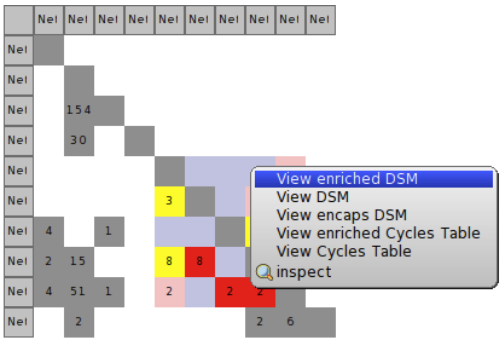

Fig. 5. Contextual menu.

The use of double-click is possible on header and on SCC. For the header, it opens a DSM with all packages on which the selected package depends. And for the SCC, it opens a window with an eDSM of the SCC, as in Figure 6.

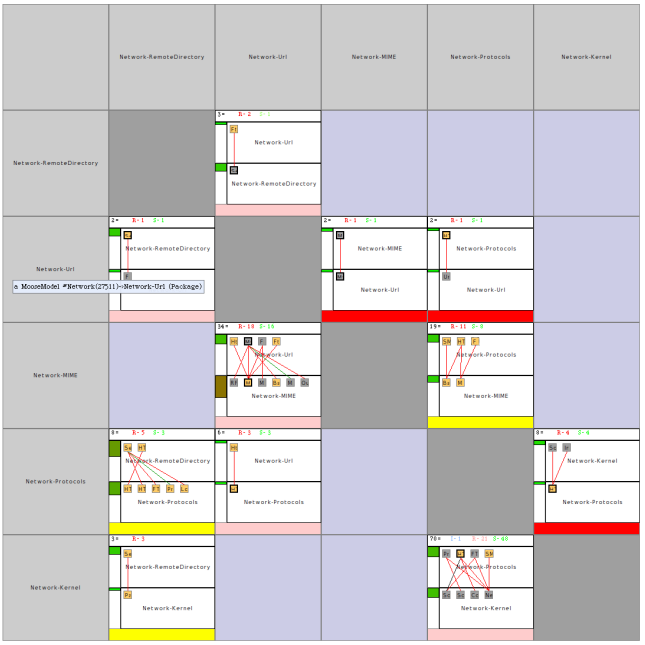

Fig. 6. one SCC centric eDSM.

3.3 Show source code

To see the source-code, it is possible to right-click on a class and to select “Browse” on the class or on a specific method as in Figure 7.

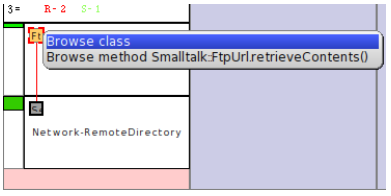

Fig. 7. We can browse source code.