

Reflective Software Engineering

Module 05:

Size Measurement Concepts

(1)



Module Objectives

Understand why size measurement is useful

Understand what hierarchical, multiple, and incremental size measures are

Understand how to enter size measurement data in Leap.

(2)



Motivation for size measurement

Provides insight into progress toward completion of a work product.

Provides insight into development rate.

Provides insight into density of a specific attribute within a work product.

(3)



Size measure characteristics

Three useful characteristics of size measurement representation for reflective software engineering are:

- hierarchical
- multiple
- incremental

Let's look at each of these next

(4)



Measurement Characteristics: Hierarchical

Hierarchical size measures provide a hierarchy of "grain sizes" for viewing a measurement.

Examples:

- inches, yards, feet
- lines, methods, classes, packages
- lines, paragraphs, sections, chapters

Each measured object at one grain size can be uniquely associated with a measured object at the next higher grain size.

(5)



Measurement Characteristics: Multiple

The size of certain objects can also be measured by multiple, (more or less) independent measures.

Examples:

- height, weight, volume
- lines of code, function points

(6)



Measurement characteristics: Incremental

An "incremental" size measure allows you to perform a "difference" between two versions of a work product and determine how much "new size" has been added.

(7)



Size measures in Leap

Size measures should support:

- hierarchical
- incremental
- multiple

Leap's size measurement representations are designed into:

- Aama (size measure definition)
- Ilio (project definition)
- Iole (size data entry)

LOCC is a tool for size measurement.

(8)



Using size: An overview

1. Define a size measure using Aama
2. Define a project using Ilio/Hee:
 - Define start and end dates
 - Define "FileList" specification for work products.
3. Enter the size data into Iole:
 - Date must fall between start/end dates
 - Path and file must match the FileList specification.
 - Use LOCC for generating Iole data for Java!
- (9) 4. Condense project and view size data.



1. Define the size measure

	Size Type	Desc/Type	First level	Second level	Third level	Fourth level	Fifth level	Sixth level
1	WordSize	TechReport	Words	Section				
2	FileSize	TechReport	Files					
3	JavaSize	Java Source	Lines	Method	Class	Package		

Buttons: Save, Load, Clear Table, Add Rows, Delete Row, Hide Rows, Show All, Close

(10)



2. Define a project using Ilio/Hee

Project	File List	Start	Stop
Project1	c:\techreports\pl.html	09:25 04/01/1999	19:41 04/01/1999

Save Load Clear Table Add Rows Delete Row... Hide Rows Show All Rows Close

Ready.

(11)

3. Enter size data into Iole

Date	Size Type	Path	File	Project	# of Units	2nd Lvl. N	3rd Lvl. N	4th Lvl. Name
04/15/1999	FileSize	c:\techreports	pl.html	Project1	332			

Save Load Clear Table Add Rows Delete Row... Hide Rows Show All Rows Close

Ready.

(12)

4. Condense and view size data

Level	Planned	Actual	Error	%
LOC	812.0	2,109.0	1,297.0	159.7
Method	85.0	91.0	6.0	7.1
Class	8.0	11.0	3.0	37.5
Package	1.0	1.0	0.0	0.0

(13)

Size data hints

Avoid manual entry of hierarchical size measures!

Formatting guidelines are often needed to ensure that size is measured consistently.

Check the Size chapter for ways to diagnose problems in the way Leap assign size entries to projects. (Note: you cannot manually enter the project field value in Iole!)

(14)

Using LOCC

LOCC is an automated tool for calculating size in a Leap-compliant fashion. It supports Java, text, and C++.

You can download the tool and obtain user documentation at:

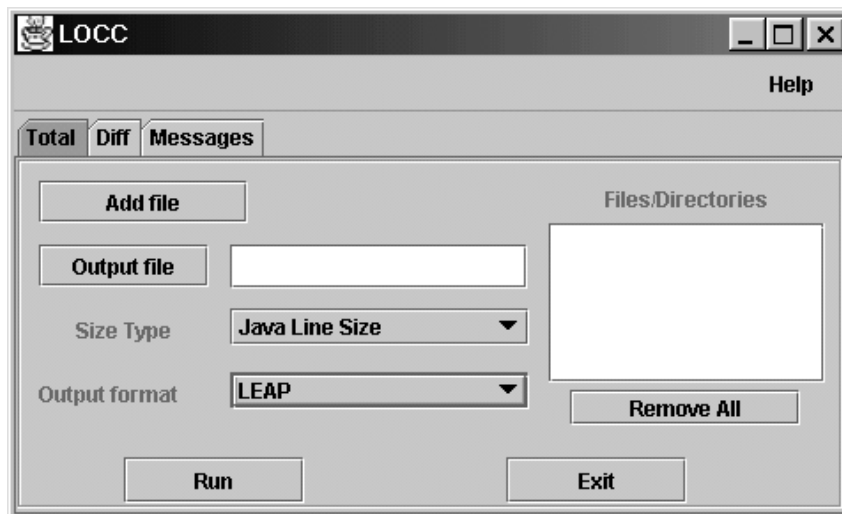
<http://csdl.ics.hawaii.edu/Tools/LOCC/LOCC.html>

You ***must*** use LOCC to count the size of your projects in this class!

(15)



The LOCC interface



(16)



Steps to using LOCC

1. Decide whether you need to compute the total size or an incremental size.
2. For total sizes, add all of the files to be counted.
3. Specify an output file and the "LEAP" format.
4. "Import" the resulting file into your Leap data file (I.e. 613-data.leap). Make sure the entries are associated with the right project.

(17)



End of Module

(18)

