

Метрика за информационния поток

(Information Flow Metrics)

Метриката за информационния поток представлява статична метрика, предложена от **Henry** и **Kafura**, която предоставя мярка за междумодулната сложност на първичния код. В основата си метриката за информационния поток се крепи върху следната концепция:

Дори и най-простата система се състои от компоненти, като акцентът тук не пада върху самите компоненти като статични единици, а върху работата, която те извършват, разглеждана като динамична структура, както и какъв е начинът, по който тези компоненти са свързани помежду си. Именно с това уточнение се дава ясна дефиниция за това- кое е определящо за идентифицирането на сложността (**complexity**) на една система.

Друго разяснение за идентифицирането на сложността:

В основата си Метриката за информационния поток се основава на следната концепция: И най-простата система се състои от компоненти (**components**). Това, което идентифицира сложността (**complexity**) на системата е работата, която тези компоненти извършват (това- до каква степен компонентът извършва една-единствена функция, определяме именно като **cohesion**), и това- до каква степен те са свързани помежду си (**coupling**).

Следните са основните термини, използвани в информационния поток:

Компонент (Component): всеки елемент, който е идентифициран при декомпозицията на една софтуерна система (разбиването ѝ на нейните съставни части).

Кохезия (Cohesion): Степента, до която даден компонент изпълнява една-единствена функция.

Каплинг (Coupling): Степента, до която един компонент е свързан с останалите компоненти в рамките на една и съща система.

Метриката за информационния поток се справя с този тип сложност чрез наблюдение и измерване на потока от информация сред компонентите и модулите на дадена система. Както споменах и в началото на статията, тази метрика ни е известна благодарение на **Henry** и **Kafura**, затова и е позната още като **Henry and Kafura's Metric**.

Тази метрика е "чувствителна" към сложността благодарение на взаимосвързаността, която се наблюдава сред отделните компоненти на системата. Измерването на потока от информация включва сложността на софтуерния модул, чийто поотк от информация бива наблюдаван, и се дефинира като сума от сложностите на отделните процедури, включени в съответния модул.

Процесът на измерването на информационния поток на даден модул допринася за сложността чрез следните два фактора:

1. Сложността на самия процедурен код.
2. Сложността, идваща от свързаността на процедурата със средата. Ефектът от първия фактор се включва към сложността чрез т.нар. **LOC (Line Of Code) measure**.

За да се осъществи количествено измерване на втория фактор, **Henry** и **Kafura** са дефинирали два термина, познати като **FAN-IN** и **FAN-OUT**.

FAN-IN: FAN-IN на една процедура се нарича броят локални потоци **КЪМ** тази процедура, като към него се прибавя и количеството структури от данни, от които дадената процедура извлича информация.

FAN-OUT: FAN-OUT на една процедура се нарича броят на локалните потоци **ОТ** тази процедура, като към него се прибавя и количеството структури от данни, които тази процедура обновява чрез произлизащите от нея потоци.

Забележка:

локални потоци **КЪМ** процедура- връзките от една процедура **КЪМ** дадената процедура.

локални потоци **ОТ** процедура- връзките **ОТ** дадената процедура към друга процедура.

Това чрез фигура 1 може да се представи по следния начин:

Процедурите са отделните възли на насочения граф (Nodes), а локалните потоци представляват ребрата (Edges), свързващи възлите.

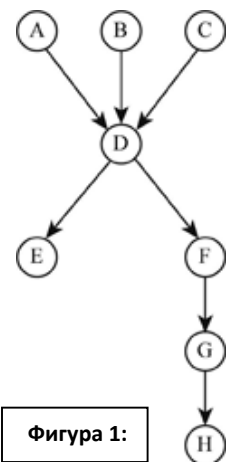
Henry and Kafura's complexity value

$$\text{PROCEDURE COMPLEXITY} = \text{LENGTH} * (\text{FAN-IN} * \text{FAN-OUT})^2 ,$$

където LENGTH е дължината на процедурата.

Забележка:

Всяка мярка за размер може да бъде използвана за изчисляване на дължината на дадена процедура. В случая за изчисляването ѝ **Henry** и **Kafura** използват именно **LoC (Line of Code) measure**.



Аргументи ЗА/ПРОТИВ описаната от мен метрика

ЗА: Описаната от мен метрика е полезна, защото предоставя възможността количествено да се измери сложността както на постъпващия в дадена система, така и на произлизащия от нея поток от информация.

ПРОТИВ: Лично за себе си не смятам, че всички необходими параметри за изчисляването на тази сложност биха могли да се конкретизират с абсолютна точност, което, от своя страна, означава, че количествената мярка не би била дотам достоверна.

Източници:

<https://www.javatpoint.com/software-engineering-information-flow-metrics>

http://uroci.karadev.net/pdf_basic/softuerni_tehnologii_kst.pdf

https://en.wikipedia.org/wiki/Programming_complexity

https://www.researchgate.net/figure/Control-flow-graph-Henry-and-Kafura-HK81-proposed-an-information-flow-complexity_fig4_3187765

<https://www.chegg.com/homework-help/definitions/henry-kafura-information-flow-3>

Изготвил:

Теодора Иванова, фак.62250

Софтуерно инженерство, 2 курс, 1 група