

Boundaries, Boundaries Everywhere!

2011

Thomas Cagley, Vice President
t.cagley@davidconsultinggroup.com

David Consulting Group
Liberty Square, Suite B-2
270 W Lancaster Ave
Malvern PA 19355
(440) 668-5717



The Problem . . .

- Many definitions although lots of similarities
- Individual perception can and does impact the definition of boundaries
- Boundary definition can impact the usefulness of the information a count generates

Your IFPUG Counting Practice Manual Definition

- An application is a cohesive collection of automated procedures and data supporting a business objective; it consists of one or more components, modules, or subsystems.

The Definition Is Refined By . . .

- Defines what is external to the application
- Is the conceptual interface between the 'internal' application and the 'external' user world
- Acts as a 'membrane' through which data processed by transactions (EIs, EOs and EQs) pass into and out from the application
- Encloses the logical data maintained by the application (ILFs)
- Assists in identifying the logical data referenced by but not maintained within this application (EIFs)
- Is dependent on the user's external business view of the application. It is independent of the technical and/or implementation considerations.



IFPUG Rules

- The following rules must apply for boundaries:
 - The boundary is determined based on the user's view. The focus is on what the user can understand and describe.
 - The boundary between related applications is based on separate functional areas as seen by the user, not on technical considerations.
 - The initial boundary already established for the application or applications being modified are not influenced by the counting scope.





Cosmic 3.0

DEFINITION – Boundary

- The boundary is defined as a conceptual interface between the software being measured and its functional users.
- NOTE: The boundary of a piece of software is the conceptual frontier between this piece and the environment in which it operates, as it is perceived externally from the perspective of its functional users. The boundary allows the measurer to distinguish, without ambiguity, what is included inside the measured software from what is part of the measured software's operating environment.



Cosmic Boundary Rules

The following rules might be useful to confirm the status of a candidate boundary:

RULES – Boundary

- a) Identify the functional user(s) that interact with the software being measured. The boundary lies between the functional users and this software.
- b) By definition, there is a boundary between each identified pair of layers where the software in one layer is the functional user of software in another, and the latter is to be measured. Similarly, there is a boundary between any two peer components in the same layer; in this case each component can be a functional user of its peer.



More Cosmic Boundary Notes

- The boundary allows a clear distinction to be made between anything that is part of the piece of software being measured (i.e. that is on the software side of the boundary) and anything that is part of the functional users' environment (i.e. that is on the functional users' side the boundary). Persistent storage is not considered as a user of the software and is therefore on the software side of the boundary.



How about MK II

Rule 1 Boundary

- 1.1 Mk II FPA is used to measure the size of the functionality required by the users of an application, within a boundary defined for the purpose of the FP count.
- 1.2 The application or part of the application enclosed by the boundary must be a **coherent body of functionality**, comprising one or more complete Logical Transaction Types. (In the following, 'Type' is dropped for ease of reading.)
- When the application boundary is drawn, it defines the conceptual border between the software and the 'user' (see further below). 'Input data' from the user crosses the boundary to enter the application. 'Output data' leaves the application and crosses the boundary to reach the user. Within the boundary are the Data Entity Types (hereinafter referred to as 'entities', for simplicity) that are processed by the Logical Transaction Types (hereinafter referred to as 'logical transactions' or just 'transactions') of the application.



Finally NESMA and IEEE

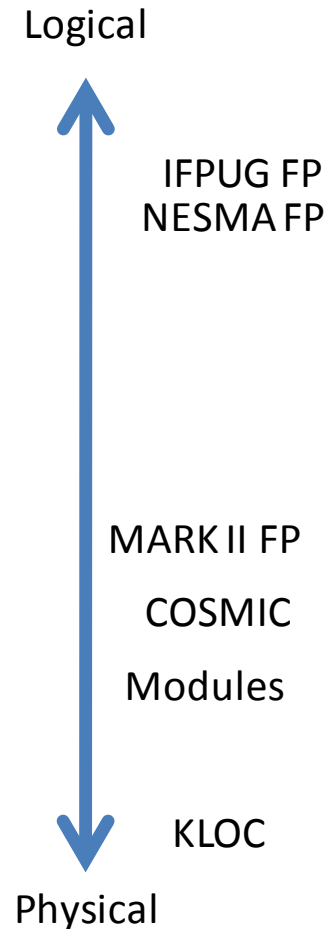
- **NESMA Application boundary:** the border between the application and its environment; i.e., other applications and users. The application boundary determines the scope for the function point count.
- **IEEE Application Boundary. (1)** the border between the application and its environment of other applications and users (*ISO/IEC 24570:2005 Software engineering -- NESMA functional size measurement method version 2.1 -- Definitions and counting guidelines for the application of Function Point Analysis*) **(2)** the border between the software being measured and the user (*ISO/IEC 20926:2003 Software engineering -- IFPUG 4.1 Unadjusted functional size measurement method -- Counting practices manual*)



Threads

Common

- Application is cohesive group of functionality and data
- Boundary defines what is inside the application and what is outside
- Differences (Glaring)
 - Boundaries drawn between layers (Cosmic)





Interpretations and Needs

- Interpretation Affect Boundaries
 - Cohesive
 - Functional Users
 - Conceptual Frontier
 - Crossing The Boundary
- Need Affects Boundaries
 - Budgeting
 - Estimation
 - Organization



An Idea . . .

- Cohesion is a qualitative measure used to measure the degree to which elements of a module are related.
- Cohesion is usually expressed as "high cohesion" or "low cohesion".
- Groups of highly cohesive modules define an application.

A Weighting Scheme

$$C(M001) = \# \text{ of dependency relationships} * 4 + \# \text{ of “related relationships”}$$

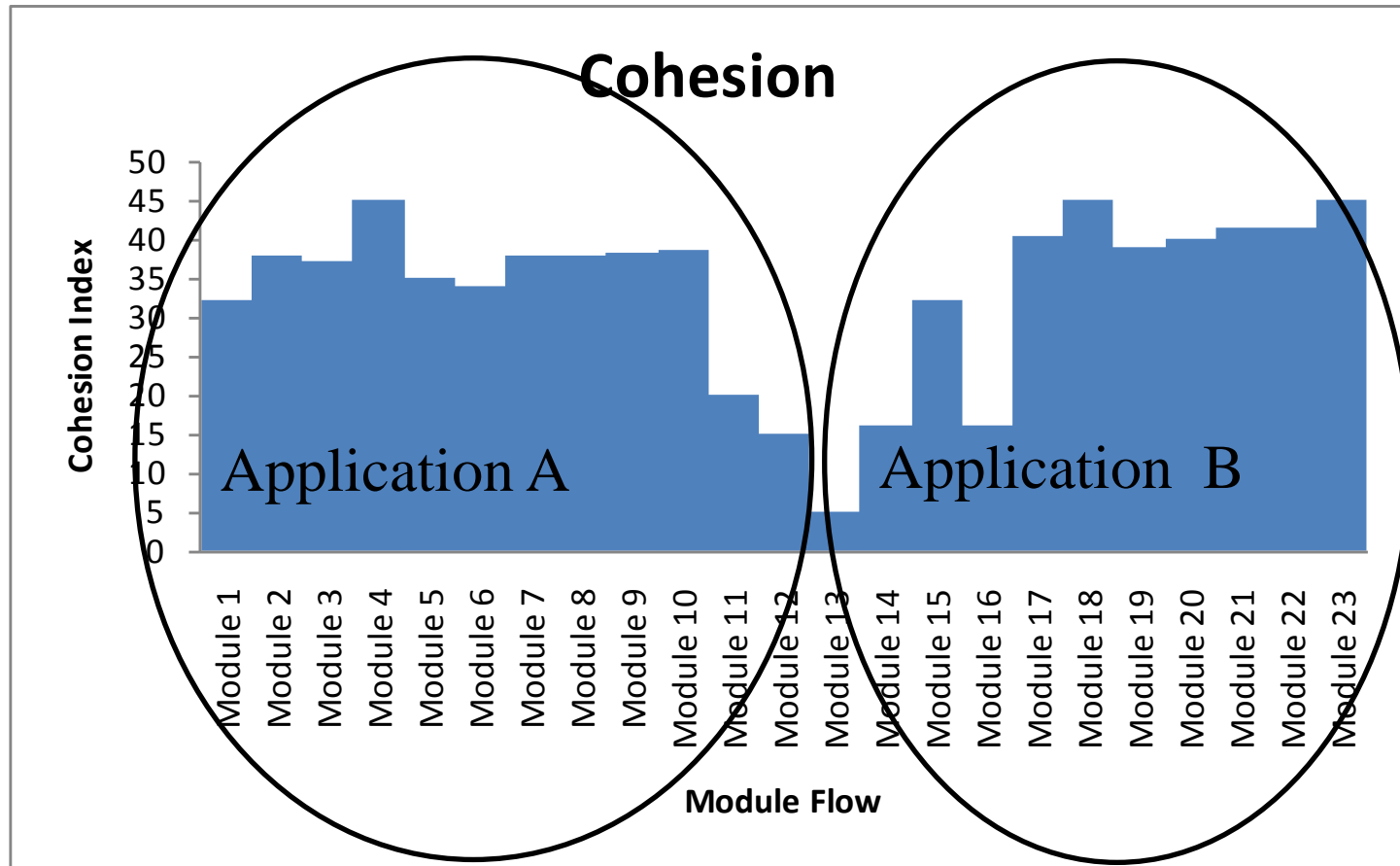
Cohesion index is calculated as following:

The # of dependency relationships is one directional and is counted for the “parent”. If module 1 is the parent of module 2, we say module 1 has one dependency and module 2 has none.

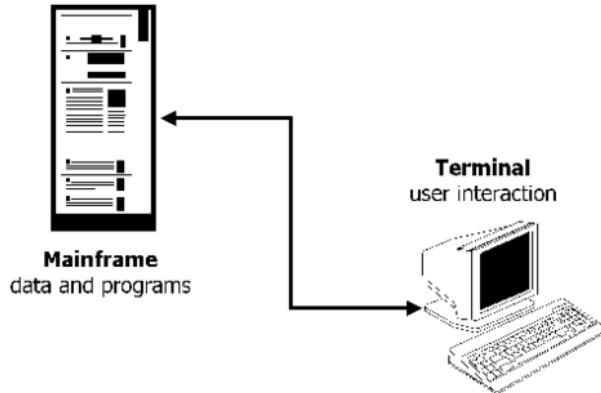
The # of “related to” relationships is counted for both modules. If module 1 is related to module 2 (“sibling”), both modules get the value. In other words, we say module 1 has one relation and module 2 has one relation.



Recognizing A Break



Classic Mainframe



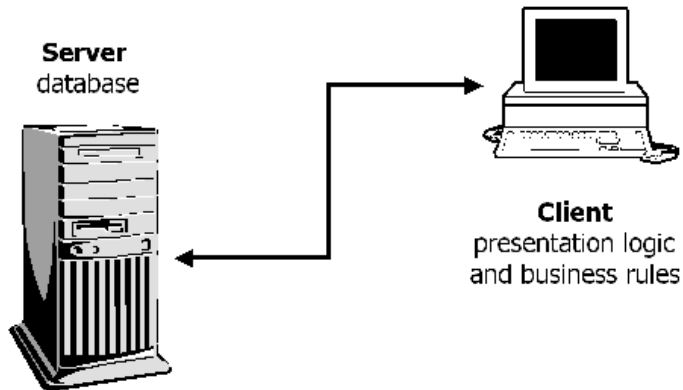
- Boundary easily recognizable all or majority of functionality resides on the mainframe.

Potential

Complications :

- How do we deal with subsystems with different functions?

Two Tier Client Server



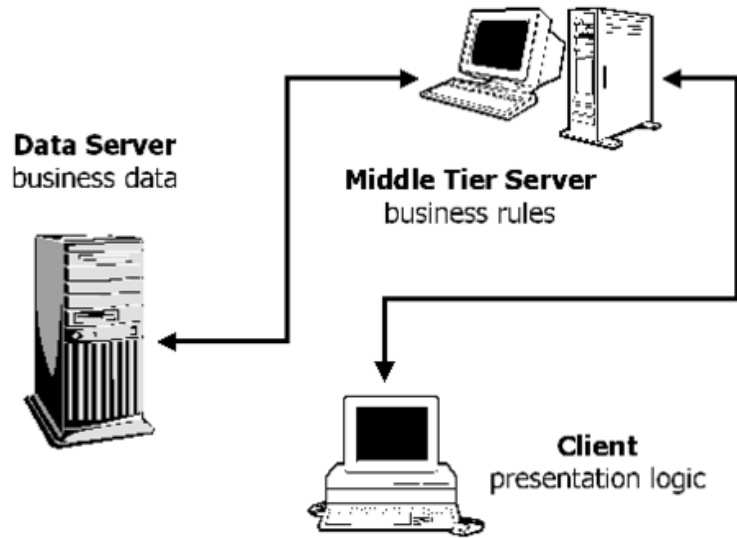
- Boundary easy to recognize.
- Cosmic helps with line “Persistent storage is not considered as a user of the software and is therefore on the software side of the boundary.”

Potential
Complications:

- What if different groups maintain the two components?



Three Tier Client Server



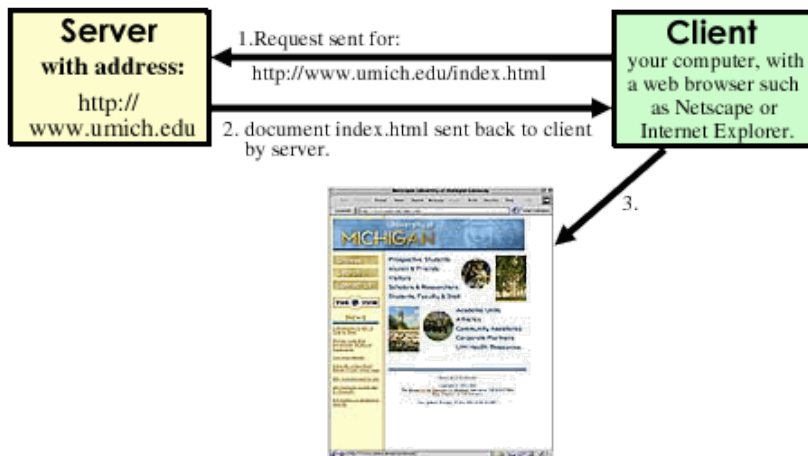
Potential Complications :

- What if the functionality is less than coherent?
- What if the components are maintain for functionality separate groups?

- Boundary easy to recognize if the functionality is related
- MKII provides supplemental guidance, “The application or part of the application enclosed by the boundary must be a coherent body of functionality”.



WEB Client Server Variant



This is what's displayed through your web browser.

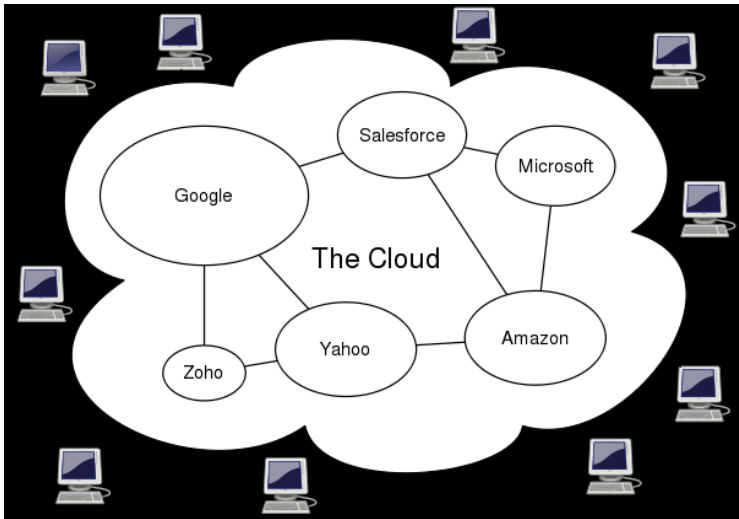
Potential Complications:

- What if the client can stand alone and operate without the WEB components?

- Does the introduction of the WEB / Browser combination change the definition of where the boundary resides?



Cloud Computing 101

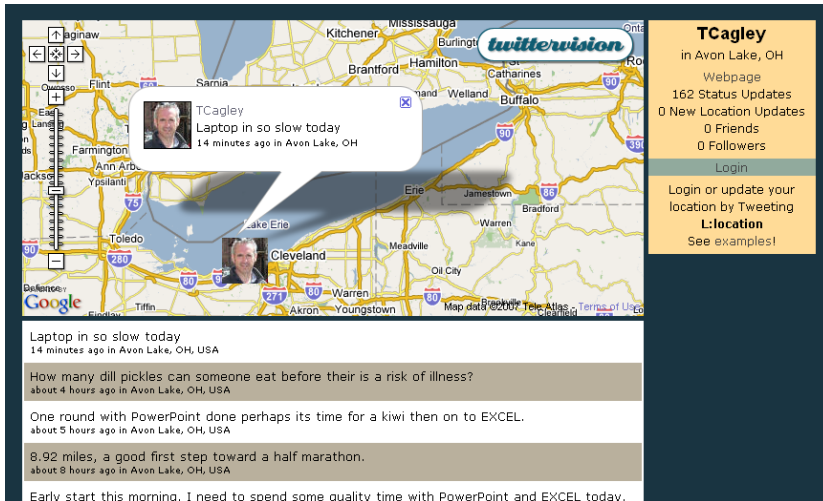


Potential Complications:

- Can one instantiation of the application on the cloud be different from another?

- Accessing applications via the cloud is as easy to recognize as any other application (example Salesforce.com)

Cloud Computing With Mashups



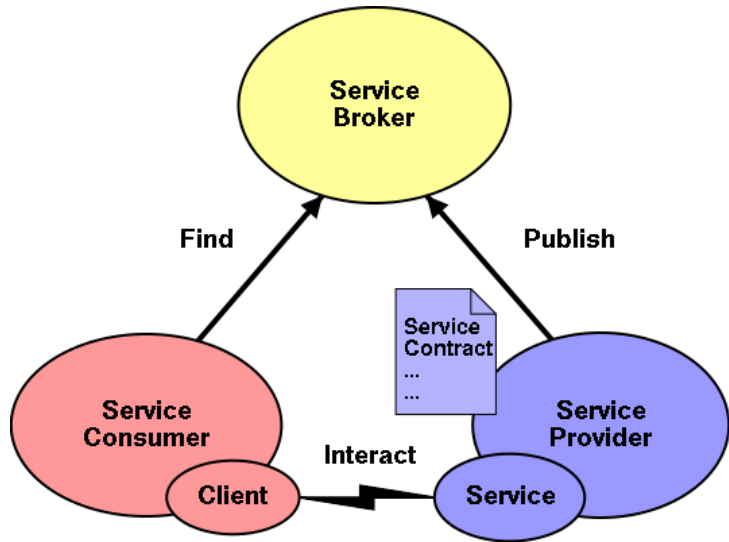
The screenshot displays a Google Maps interface with a 'twittervision' overlay. A tweet from 'TCagley' is highlighted, stating 'Laptop in so slow today' posted 14 minutes ago in Avon Lake, OH. The sidebar on the right shows the user's profile: 'TCagley in Avon Lake, OH', 'Webpage', '162 Status Updates', '0 New Location Updates', '0 Friends', and '0 Followers'. Below the profile are 'Login' and 'Login or update your location by Tweeting' buttons, along with a 'Location' field and a 'See examples!' link. The map shows the area around Avon Lake, OH, with various cities and roads visible.

- Mashups combine functionality of two or more applications to create something create greater than the sum of the parts.

Potential Complications:

- What is the impact to the boundary of an application when an user can change it at will?

Service Oriented Architecture



- Functionality is delivered from an assemblage of services from many sources bought, build or borrowed.

Potential Complications:

- Services can be assembled like Lego. If you produce components do they have a size?

Places to Visit and Things to Read:

- International Function Point Users - <http://www.IFPUG.org/>
- Common Software Measurement International Consortium - <http://www.cosmicon.com/>
- Netherlands Software Metrics Association - <http://www.nesma.nl/section/home/>

Bounding the Presentation

Questions?
Answers?
Comments.....

Thomas Cagley
t.cagley@davidconsultinggroup.com
(440) 668-5717