iTip	Guidance from the Functional Sizing Standards Committee on topics important to you	

Shared Data

Real-time Requests

iTip # 05 – (Version # 1.1 *09/21/2015)*

iTips provide guidance on topics important to the FPA community. They explain the application of IFPUG FPA method in a particular situation. iTips are not rules, but interpretation of the rules, and provide guidance using a realistic example to explain the topic being covered.

This iTip is focused on describing the IFPUG FPA method as it applies to data sharing in a real-time environment from the perspective of the application requiring the data. This iTip includes a series of examples but is not an exhaustive examination of the subject. For further examples, please see the current CPM.

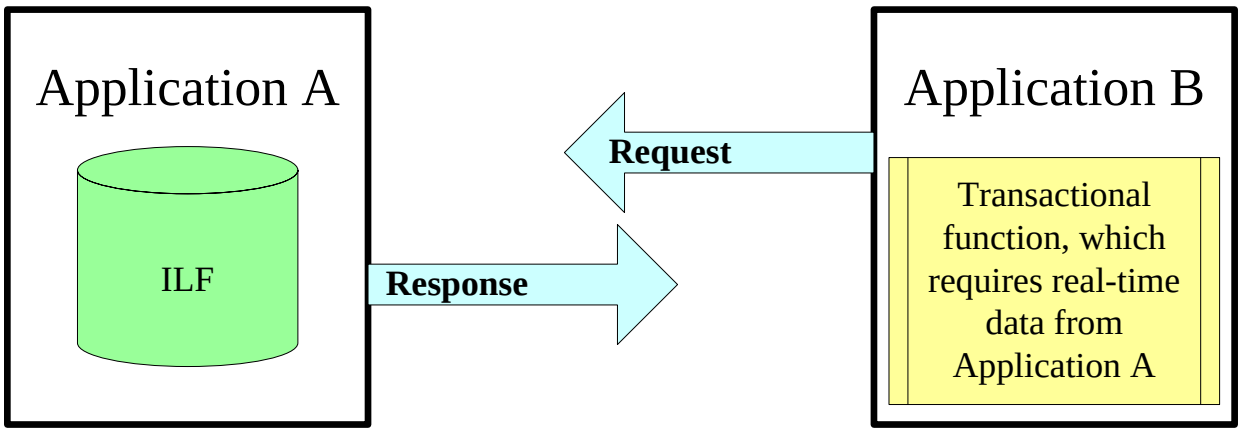
Background

Chapter 3 in Part 3 of the current CPM addresses the sharing of data between applications in a number of scenarios, but does not address the real-time environment. This iTip provides additional guidance for counting the exchange of data through APIs, stored procedures and Web Services. The examples provided focus on situations where Application B has a functional user requirement to reference data obtained from Application A to complete its elementary processes. Application B is the application being measured.

Example 1: Real-time Data Request/Response

Application B requires data from Application A to complete a real-time transactional function. Application B uses the data to complete transactional processing (e.g., to display data on a screen) in Application B.

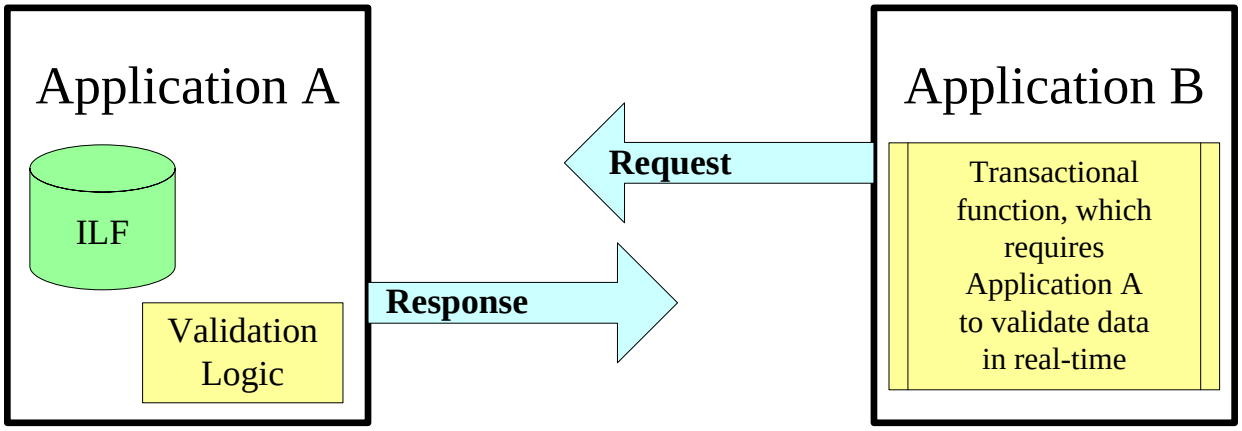
In order to obtain the required data, Application B sends a request to Application A. Application A processes the request, accesses its data and sends a response with the required data to Application B.



From Application B's perspective, a transactional function in Application B requires data from Application A. The primary intent is to reference Application A's ILF, regardless of how that is technically enabled. Application B's elementary process is not complete until all processing steps have been completed, including referencing data from Application A. For Application B the messaging between the two applications (i.e., the request data and response) is part of the elementary process of the input transaction being processed by Application B and is not counted separately. Application B counts an EIF and an additional FTR for the transactional function. The DETs and RETs are determined by Application B's logical view of the data and the attributes actually passed and returned.

Example 2: Real-time Data Validation Request/Response

Application B processes a transaction that requires Application B to validate employment information. Since Application A owns and maintains Employee Data, this is accomplished by Application B sending a request to Application A. Application A accesses its employee file and sends a response with the results of the validation to Application B. The code for the validation resides in and is maintained by Application A. Application B uses the response to complete its processing.

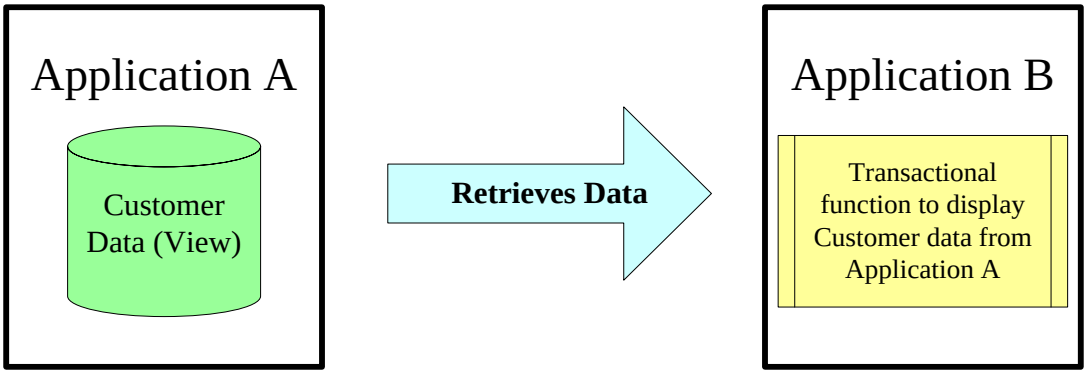


From Application B's perspective, there is a functional user requirement to process a transaction that includes validating employment using data maintained within Application A. The validation is logically performed as part of the Application B

elementary process, but is physically performed by Application A. For Application B the messaging between the two applications (i.e., the request data and response) is part of the elementary process of the input transaction being processed and is not counted separately. The primary intent is for Application B to reference Application A's data in order to complete its own elementary process, regardless of the physical implementation. Application B counts an EIF for the group of logically related data, represented by the request and response. The DETs and RETs are determined by Application B's logical view of the data and the attributes actually passed and returned. An additional FTR is also counted for the transactional function being evaluated.

Example 3: Database View Created for Reference

Application B presents data that is owned and maintained by Application A to the user in an on-line query. Application A provides a database view that Application B uses to reference the data. In the implementation of this requirement, Application A creates a database view of its Customer data filtering and summarizing the data so that Application B can reference a specific subset. This view of Application A's data is created and maintained specifically for Application B; this view is not utilized in any of Application A's transactional functions.



From Application B's perspective, there is a functional user requirement to perform an on-line query, displaying data from Application A. The primary intent is for Application B to reference Application A's data to complete its own elementary process, regardless of the physical implementation. Customer is an external interface file for Application B. The DETs and RETs are determined by Application B's logical view of the data and the attributes actually used. It is also counted as an FTR for any transaction in Application B referencing that database view.

Summary

While this iTip illustrates data sharing scenarios specific to a real-time environment, the approach is intended to be technology-independent and can be applied to many technologies and platforms. These examples translate a number of scenarios of "how data is referenced" back to the focus of "what function is provided" per the Functional User Requirement. In all cases, the counting interpretation is

based on Application B's user view and functional requirements. Analyzing the primary intent is key to that determination. In all of these examples, the primary intent for Application B is to reference data that Application A maintains. As a result, in each example an EIF is counted within the Application B boundary, regardless of "how" the data is physically referenced.

Frequently Asked Questions (FAQ)

1. What happens when multiple applications invoke a common interface to obtain or validate Application A's data?

   In this variation, each application using the common interface counts an FTR and EIF.

2. What happens when multiple transactions within the same application invoke the common interface to obtain or validate data?

   In this variation, each transaction in Application B using the common interface counts an FTR. One EIF is counted for Application B.

3. What if the data returned by Application A actually came from multiple ILFs?

   The DETs and RETs are determined by Application B's logical view of the data and the attributes actually used. If this is considered one logical group of data, it is counted as one EIF in Application B, regardless of Application A's view. If it is considered more than one logical group of data in Application B, it would be counted as more than one EIF. The CPM states that the data must be "identified in an ILF in one or more other applications". It does not stipulate that there only be one ILF to one EIF.

4. How does Application A count its response to the request?

   iTip #6 presents these scenarios from the perspective of Application A. Refer to the Resources > iTips / uTips page.

Further Reading

IFPUG Counting Practices Manual (CPM 4.3.1), Part 1, Section 5.5 – Measure Transactional Functions.

IFPUG CPM 4.3.1, Part 2, Chapter 7 – Measure Transactional Functions.

IFPUG CPM 4.3.1, Part 3, Chapter 3 – Shared Data.