



Counting Data from Another Application

Daniel Horvath

Application development does not occur in a vacuum. It is quite common that data integration requirements stipulate that data from one application be used in another application.

When conducting a function point count for a given application, the function point counter must be able to distinguish between External Input transactions where the data originates from another application, and External Interface Files which are referenced by the application being counted. In order to accomplish this, the function point counter must first be able to define and understand the boundaries between the application being counted, and other applications.

UNDERSTANDING THE BOUNDARIES

When considering data from another application, the boundary between the application being counted, and that of the other application, must first be clearly understood. The boundary defines what is internal to the application being counted, and what is external. It establishes a “membrane” through which data processed by external input (EI), output (EO), and inquiry (EQ) transactions pass.

In order to establish the boundary, the function point counter must apply the Counting Practices Manual (CPM) rules. The boundary is based on the user’s view of the application. It must be that which the user is able to understand and describe. In addition, the boundary between two different applications is based on the user’s

functional requirements and not on technical considerations.

The function point counter must therefore establish whether the applications fulfill specific, separate business requirements. Furthermore, it must be made clear which business functions (and related data) belong with which application. This is critical to the process of counting data that is from another application.

Once the boundaries are established, data from an application other than the one being counted can be examined to identify the functional component type.

RULES FOR COUNTING EI AND EIF FUNCTIONS

The purpose of this article is to clarify when to count data being received into the application as External Input transactions and when to count the data as External Interface Files (EIFs).

In order to decide whether data from another application should be counted as an EI or an EIF, the function point analysis process should be followed carefully. Once the counting scope and the boundaries are identified and understood, the next step is to count data and transaction function types. At this point, the data and transaction functions are examined in detail in order to be counted.

Data function types are logical groups of data that are required by the user to meet their business requirements. When the logical group of data is internal to the application being

counted, it is considered to be an Internal Logical File (ILF). When the logical group of data is external to the application being counted and is maintained by another application then it is counted as an External Interface File.

Transaction functions provide the user the capabilities to interact with the data maintained and referenced by their application using specific business rules that dictate how the data is processed. The processing may include data entering the application boundary for the purpose of maintaining one or more ILFs or to control the system behavior. This transaction would be counted as an External Input. Alternatively, the processing may be an elementary process to send data or control information outside of the application boundary. Depending on the specific processing logic of the transaction being evaluated it may be counted as either an External Output or External Inquiry function.

When data from another application is utilized, it may be either an EI or an EIF, depending on the specific logic of the elementary process being analyzed. To help with this determination, the rules will need to be examined in even more explicit detail.

The Counting Practices Manual specifically defines an EI as an elementary process where data enters the application boundary, where the primary intent is to maintain one or more Internal Logical Files. An EI may also be counted if control information enters the boundary. The CPM also specifically defines an EIF as a group of logically related data referenced by the application, but maintained within the boundary of another application (i.e. the group of data must be an Internal Logical File in another application). The primary intent is to hold data referenced by elementary processes within the application being counted.

By applying these definitions:

- Data entering the application boundary for the purpose of maintaining an ILF, is identified as an EI
- Data that is only referenced by an elementary process in the application being counted, is an EIF

HINTS ON HOW TO DECIDE BETWEEN EI AND EIF FUNCTIONS

Once the rules regarding the counting of EI and EIF functions are understood, they can be applied to specific, real-world cases. In order to assist with the application of the rules in deciding between EIs and EIFs, listed below are a set of questions that should be considered. The answer to any single one of the questions may not provide the complete answer about how to count the function; others may need to be asked as well. The questions are intended to be thought-starters only.

- Assuming that the boundaries are well understood, is it clear that the data being counted is, in fact, from another application? If it isn't, it cannot be an EIF.
- Are the requirements completely and clearly documented regarding how the data from the other application is to be used? Without clear documentation, deciding on how to count these functions becomes difficult to impossible.
- Is the data received by or does it enter the boundary of the application being counted? If not, it cannot be an EI or an EIF.
- Is the data from the other application used only as reference by the application being counted? If the data does not enter the boundary with the intent to maintain one or more ILFs or alter the behavior of the application, it cannot be an EI.
- Is the data from the other application used for more than one purpose by the application being counted? It is possible that the data can be counted as both an

EI and an EIF, provided that all the criteria for both functions are met.

EXAMPLES OF COUNTING EI AND EIF FUNCTIONS

Here are some specific cases to consider:

- Data maintained in another application (an ILF in that application) is read by the application being counted and enters the application boundary as a set of transactions for the purpose of updating an ILF. This should be counted as an External Input. Make sure to look for unique External Inputs within the transaction file.
- Data maintained in another application (an ILF in that application) is used as reference by any EI, EQ or EO transaction in the application being counted. This should be counted as an EIF function, and it should be included as a File Type Referenced (FTR) for the transaction(s) that use it.
- Data maintained in another application (an ILF in that application) is loaded directly into a file or table in the application being counted, and is **not** maintained by any other transaction in the application being counted. This is often done due to technical incompatibilities or for performance reasons. In this case, the load from the one application to the other is a technical solution – the business requirement was simply to *reference* the data in the application being counted. This should be counted as an EIF function, and it should be included as an FTR for the transaction(s) that use it.
- Data maintained in another application (an ILF in that application) is read by the application being counted and enters the application boundary for the purpose of updating an ILF. The same data is *also* used for reference by a *different* EI, EQ or EO transaction of the application being counted. The data element(s) used for updating the ILF and the one(s) used for reference may or may not be the same. This should be counted as an EI transaction for the data crossing the boundary and updating the ILF, and should include a count of only the DETS that are used for the update. An EIF should also be counted for the data being used as reference and the EIF should be included as an FTR for the transaction(s) that use it. The DETs counted for the EIF should only be those fields that are used for reference.
- Data maintained in another application (an ILF in that application) is read by the application being counted and enters the application boundary for the purpose of updating an ILF. The same data is also used for reference during the same elementary process. The data should be counted as an EI only.
- A regularly scheduled batch job causes an ILF to be updated. As part of the process data from another application is referenced and is used in a calculation. The file containing the data being used should be counted as an EIF to the application being counted.

CONCLUSION

It is of utmost importance that the Function Point Counting Rules as defined in the CPM be adhered to when considering data from an application other than the one being counted. The most important considerations are the boundaries between the applications and the way the data is used by the application being counted.

Simply put, when the data from another application enters the boundary of the application being counted for the purpose of maintaining an Internal Logical File directly, it is an External Input. When the data from the other application is used as reference by a transaction within the application being counted, it is an External Interface File. Understanding

the underlying business process requiring the data is the ultimate key to counting it correctly.

About the author

Daniel Horvath, Senior Management Consultant, Q/P Management Group wrote this article. Mr. Horvath specializes in project management and software engineering metrics, including function point analysis.