

Different applications, different views of the data

Carlos Eduardo Vazquez¹

Abstract

The CPM is absolutely clear that different applications with different views from the data requirements have different analysis when it comes to determining data functions' complexity; each one can have a different number of DET and the information counted as one DET for an ILF/EIF in one application may be counted as more than one DET in another application. However, when CPM establishes that an EIF must be accounted as an ILF in another application, it does not clarify if it must be one and only one ILF.

For this reason, the iTip #5 – Shared Data Real-Time Requests, published by IFPUG, provided additional guidance to apply IFPUG FPA method to data sharing in a real-time environment from the perspective of the application requiring the data; scenarios typically related to this topic use Web Services, relational databases' Stored Procedures or Views. iTip #5 includes a series of examples but is not an exhaustive examination of the subject. Thus, this document provides an additional illustration of the FAQ presented in iTip #5.

Whenever the functional requirements specification is not available or it does not fulfill its purpose of capturing how the user views the information flow through the application, it helps to extrapolate the requirements from the design.

Keywords

software measurement, web services measurement; FPA; IFPUG; shared data measurement; relational data base views measurement; relational data base stored procedures measurement; exported subroutines measurement

¹FATTO Consulting, carlos.vazquez@fattocs.com

Contents

1 Problem	1
1.1 Example	1
2 Analysis	2
3 Design/Implementation Constructs	2
4 Counter examples	2
4.1 Another Counter example	2

1. Problem

There are functional requirements of **Application A** (depicted by EI and EO in figure 01). Within the description of its steps, there is one in particular which gets a **data set D** from logical files from another **application B**. **Application A** users' view sees **data set D** as a unique logically related data group.

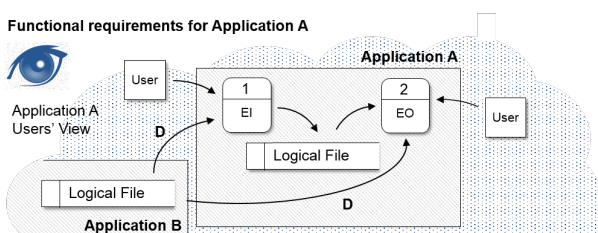


Figure 1. functional requirements model of Application A .

Application B spans some of D data items over different log-

ical files. Those logical files maintain additional data items of interest only of **Application B**. **Application B** derives the data items within D not maintained within those logical as described by its functional requirements model depicted by the EO in figure 02. The **Application B** user view dictates the identification of different logical files regardless of its implementation.

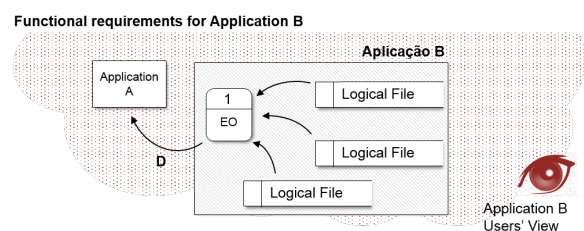


Figure 2. functional requirements model of Application B .

1.1 Example

As part of a banking loan process, the **Registration Application** (GECAD) requires information about **Agreements** regulating individuals and organizations within an **Economic Group** in order to fulfill the **Issue Registration Report** process for a specific economic group.

In the GECAD users' view, the Balance Due data item regarding past agreements is functionally dependent from information about agreements. The GECAD requires a subset of this information (depicted as **obligation** in figure

3) in order to issue a registration report. GECAD user does not know and he does not have the responsibility to define or specify the business rules that apply in order to compute the balance due. The knowledge and responsibility for it lies within the **Contract Management Application** (GEFIN) domain.

The event GECAD Application must respond to is the **registration report request**. It is not the event of a new **obligation** sent from GEFIN Application.

Functional requirements for GECAD

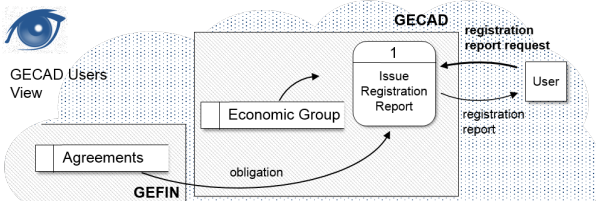


Figure 3. functional requirements model of GECAD.

The GEFIN comprises, as part of its functional requirements, a **View Integrated Agreement Summary** process to deliver an integrated view of agreement data from a series of different logical files, each of which maintain different aspects of agreement data according to its users base; for instance, funding request data, funding parameters data, financial plan data, etc. GEFIN users have identified this requirement because many are the applications needing different subsets of this data as an integrated view describing a contract. Figure 4 depicts this scenario.

Functional requirements for GEFIN

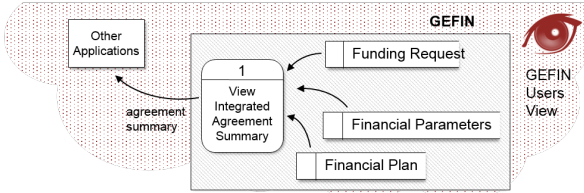


Figure 4. functional requirements model of Application GEFIN .

2. Analysis

An EIF must be measured in **Application A** and an EO/EQ should be measured in **Application B**; In the scenario described, it is the case for an EO because there is processing logic for mathematical formulas and calculations regarding the Balance Due among the business rules that apply. Otherwise, if there were none of the four required processing logic types to segregate an EO from an EQ, an EQ should be identified instead.

The definition of EIF in CPM requires the data to be maintained as an ILF in another application. The rule does not specify if it is one and only one ILF in other application. However, the FSSC has provided further guidance on this topic in iTip #5 - Shared Data, Real-Time Requests: "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".

Different applications, different views of the data — 2/2

As for different visions regarding data items, the CPM is already quite clear about the matter. It determines that the two different applications should account for different amounts of DET according to different users' views.

3. Design/Implementation Constructs

The design or implementations of those requirements may use:

- Web Services;
- Relational data base Views;
- Relational data base Stored Procedures;
- Subroutines exported for external users;

Those items do not intend to be a complete list of possible design or implementation scenarios and each platform or technology has its own resources to comply to those same requirements.

4. Counter examples

Application A needs to get data from **application B** in order to perform some validation in data entries, complement information of its own related to information maintained within **Application B**. **Application B** provides the data by a Web Service or a relational database View. There is no business rule enforced by this construct and the reason why the implementation of these **Application A** requirements are implemented in **Application B** in such a way is because of a corporate IT policy preventing direct access to whatever application data by another application.

If there were no such policy, then there would be nothing to prevent **Application A** from referencing data from **Application B**, without **Application B** having an action or reaction.

4.1 Another Counter example

Application A responds to an event initiated by **Application B** as described in "Scenario 8: Standard Transaction Data".