Software Reuse Evaluation based on Functional Similarity in COSMIC-FFP Size Components

Luca Santillo, Alain Abran

Abstract

Software reuse is often recommended for improving the productivity of the development process. However, recognizing opportunities for reuse remains a challenge. This work proposes a technique to identify opportunities for reuse based on the similarity between software functions. This technique, referred to here as "functional similarity", is based on functional information collected by the COSMIC-FFP measurement method during the measurement of the software. The proposed approach is applied to a set of measurement case studies for which opportunities for functional reuse have been identified and quantified.

1. Introduction

The size of a software system is considered to be its main cost driver: a number of authors have documented, through statistical analysis, a positive relationship between size and development effort, even though "size" information alone does not explain its full relationship to effort. Software reuse can also significantly impact project effort, as can other relevant factors, such as, for instance, software complexity and the number of change requests made throughout the project process.

In this paper, an approach is proposed to evaluate the potential for reuse based on early project information obtained during the functional requirements measurement phase. The proposal is aimed at finding opportunities for reuse using functional elements identified when measuring the functional size of a software project, and is based on uncovering similarities between functional processes measured using the COSMIC-FFP sizing method, taking into account the data movements and data manipulations. This proposed approach is illustrated with a set of eight case studies of functional requirements measured with the COSMIC-FFP ISO 19761 standard [1].

The motivation for the development of such a technique, and its underlying model, is to lower development cost, while at the same time improving the estimation of development and enhancement effort.

The paper is organized as follows. Section 2 presents the proposed approach to finding functional reuse opportunities. Section 3 presents the empirical data set available to illustrate an application of this functional similarity-based approach. Section 4 presents the similarity assessment results, and section 5 a summary and a discussion.

2. The Functional Similarity Assessment Approach

2.1. Context

Any software system can be modeled as a hierarchy, the whole-system level being decomposed into one or more independently acting "modules". Every module is made up of a set of functional processes, and each functional process has a number of base functional components. In the COSMIC-FFP measurement method, modules typically correspond to software peer items belonging to a particular layer (the layer concept should be considered orthogonal to the hierarchical structure; that is, each layer can be considered an independent system in and of itself, for measurement purposes).

This paper proposes a technique for finding opportunities for reuse based on the similarity of software functions in any software modeled as a hierarchy. A functional similarity assessment can be based on the underlying structure of the components of the software being examined and measured, and can be derived as a rating of the *similarity* concept, defined as follows:

Two functions are considered similar if they can be decomposed into the same subset of data movements and/or data manipulations.

Such an assessment can be made according to various orders of approximation, from a direct comparison – using no details – to progressively more detailed comparisons. The lowest, or zero-order level of assessment, would be performed based only on human-based comparison of functional process descriptions. At higher levels, the assessment would use progressively more details of functional process measurements, such as the amount and classification of their data movements and data processes, as suggested by the similarity concept defined above.

Since the COSMIC-FFP measurement method defines its base functional components as data movement types (Entries – E, eXits – X, Reads – R, Writes – W), and each data movement must refer to a single data group [2], it is easy to base the similarity value of one functional process with respect to another, at the first order of assessment, by analyzing the COSMIC-FFP measurement results. In addition, since the COSMIC-FFP measurement method allows for local extensions to include further details in their own measurement details, a simple classification of data manipulation types is proposed to help compare functional processes based on the amount of data movements and/or of data manipulations they have in common.

2.2. First-order evaluation: Data movements only

Comparison between functional processes in terms of data movements only (e.g. "shared DMs") is proposed as a first-order evaluation of functional similarity, since it is easy to obtain such information straight from the measurement records. The numerical similarity values in this assessment are assigned on the basis of the percentages of data movements considered as similar across the functional processes being compared; for instance, if functional processes A and B are made up of data movements A1, A2, A3, A4, A5, and B1, B2, B3, and if it is verified that A1-B1, A2-B2 and A3-B3 are the same (i.e. the same data movement type over the same data group), then the similarity values for A and B would be the following:

A: 100% similar to B (i.e. A makes use of all the same data movements as B);

B: 60% similar to A (i.e. B makes use of 3 out of 5, or 60%, of the data movements of A).

While exact percentages can be derived by carrying out a similarity assessment for each pair of functional processes, similarity values are proposed in Table 1, for purposes of illustration, for five numerical ranges which have been selected and discretized; these ranges are based, by analogy and adaptation, on the enhancement adjustment of NESMA function points [3].

Table 1. Similarity matrix for a functional process – 1st order (discretized).

Shared DMs	Null (<10%)	Low (10-30%)	Avg (30-70%)	High (70-95%)	Max (>95%)
Similarity Value	0% (entirely different)	20%	50%	80%	100% (nearly identical)

2.3. Second-order evaluation: Data movements and data manipulation

The second order of the functional similarity evaluation technique is defined to take into account both data movement and data manipulation action types, as illustrated in the form of a matrix (Table 2). From a conceptual perspective, the matrix format considers that the data movements and data manipulation are orthogonal dimensions which cannot be added up or compared with one another (e.g. the same data movement types, but with different data manipulations, or vice-versa). Also, in the current proposal, the matrix is symmetrical with respect to its diagonal, i.e. the percentage amounts of shared data movements and data manipulations are taken as having the same impact on similarity; this statement could be further refined whenever different weights in an application should, in practice, be highlighted. Moreover, further improvements to the proposed approach could be derived by extending the similarity matrix with more percentage ranges, or by requiring that the comparison of "same actions" be performed over one data group at a time, that is, at the data movement level, rather than only at the functional process level.

In Table 2, the data movements are referred to as "DMov", while the data manipulations are referred to as "DMan".

Table 2. Similarity matrix for a functional process – 2nd order.

			Shared DMovs		
	Null (<10%)	**Low (10-30%)**	**Avg (30-70%)**	**High (70-95%)**	**Max (>95%)**
Shared DMans	-	-	-	-	-
Null (<10%)	0%	5%	10%	20%	40%
Low (10-30%)	5%	20%	30%	40%	50%
Avg (30-70%)	10%	30%	50%	60%	70%
High (70-95%)	20%	40%	60%	80%	90%
Max (>95%)	40%	50%	70%	90%	100%

The second-order similarity assessment technique is thus based on the data manipulation part of a function, along with the data movements already taken into account in the previous order. The data manipulation part of a function is not currently taken into account in the COSMIC-FFP ISO 19761 standard. However, it is specifically stated in this standard that local measurement extensions can be defined to handle situations that are deemed important in some particular context. Through this local extension, the set of data manipulation primitive actions could be used to compare, at a more detailed level, the functional processes in the software being examined, by comparing whether or not they perform the same action (i.e. data usage). Table 3 lists a set of actions that can be performed by a functional process, along with a simple classification and abbreviation scheme; this list is an adaptation of an action-type list from first generation functional sizing methods [4]. It is to be noted that actions numbered 1 through 4, in the proposed sequence, must necessarily be identified as equivalent to data movements, and thus do not contribute to the data manipulation count.

Table 3. Action-type list.

No.	Action	COSMIC-FFP Function Types
1	Data acceptance from outside the system's boundary	Data Movement (Entry-type, E)
2	Data presentation outside the system's boundary	Data Movement (eXit-type, X)
3	Data group reference/retrieval (read)	Data Movement (Read-type, R)
4	Data group insert/update (write)	Data Movement (Write-type, W)
5	Derived data creation by transforming existing data	Data Manipulation (creation, D)
6	Mathematical formulas/calculations	Data Manipulation (creation, M)
7	Condition analysis to determine which are applicable	Data Manipulation (check, A)
8	Data validation	Data Manipulation (check, V)
9	Equivalent-value conversion	Data Manipulation (check, C)
10	Data filtering/selection by specified criteria	Data Manipulation (check, F)

Further explanations or remarks on how the proposed assessment is applied in practice can be found in section 4, along with some empirical results.

3. The empirical data set available

A research initiative was begun in 2005 to develop a set of software functional size measurement standard "etalons" with the COSMIC-FFP method [5, 6]. The proposed initial set contains eight case studies documenting, in a standardized format, the COSMIC-FFP functional size of five sets of requirements from ISO 14143-4, two from the RUP training material available on the Web, in addition to the Rice Cooke case study. The documentation of each case study contains the requirements documents, the corresponding UML Use Cases and the measurement results in COSMIC-FFP functional size units (Cfsu).

Table 4 provides the number of functional processes, in addition to their size, for each of these eight COSMIC-FFP case studies. For each of them, the verification level is also reported, on a scale from A (minimum) to F (maximum), where A means "verified by the measurer himself", D means verified at the highest level, E means verified by an ISO Working Group, and, finally, F means that it (would have?) been accepted as an ISO International Standard. Most of these case studies have been verified at the B level (verified by an independent expert) or C level (verified by a COSMIC Group project leader).

Table 4. Overview of COSMIC-FFP case studies available

No.	Software System	Reference Document	Functional Processes	Size (Cfsu)	Verification Level
1	Automatic Line Switching (ALS)	ISO 14143-4 - RUR B8	14	66	C
2	Gateway Application (SAGA)	ISO 14143-4 - RUR B10	19	117	B
3	Valve Control (VC)	ISO 14143-4 - RUR B9	1	12	C
4	Hotel Reservation System (HRS)	ISO 14143-4 - RUR A1	7	66	C
5	L-Euchre System (LES)	ISO 14143-4 - RUR B11	15	61	B
6	Rice Cooker (RC)	Rice Cooker Requirements	3	12	D
7	Course Registration System (CRS)	CRS-RUP	19	96	C
8	Collegiate Sports Paging System (CSPS)	CSPS-RUP	27	136	B

The analysis required to identify functional similarity was carried out for each of these case studies individually. No attempt was made to identify functional similarity across case studies. A more extensive application of the functional similarity evaluation could include more case studies, based on the same software system, e.g. after a software system has been developed and installed (baseline). The functional measurement of any enhancement project for that system could highlight functions that are changed or added, with weak or strong similarity to pre-existing baseline functions.

4. Functional similarity assessment results

This section reports on the results of the evaluation of functional similarity obtained with the proposed technique applied to the empirical data set described in the previous section.

4.1. First-order evaluation results

To assess the similarity across functional processes in each case study, each functional process, out of, say, N functional processes in a given case study, has been compared with the remaining N-1 functional processes: each time two functional processes in the measurement report are identified as having the "same" data movements with the "same" data groups, they were considered to be the 'same' or 'similar'.

The way we have assigned similarity across functional processes in each case study is based on one of the following three criteria:

1. The same data movement (data movements that are the same share not only their own type and the underlying data group, but also the data portion that they actually move);
2. The same data movement 'type' (same type and same data group, but possibly slightly different subsets of data portions being moved – it would not be possible to identify such a difference from the textual descriptions of the functional requirements);
3. In some cases, where the above criteria could not be applied as is, the analyst's best judgment (heuristic interpretation of similarity, based on the apparent similarity of the measurer's descriptions of the functional processes, of their triggering events, their data movements or their data groups).

Table 5 presents the numerical results of the application of the proposed similarity assessment technique at the first-order assessment stage (i.e. comparison of data movements between functional processes within each case study, regardless of possible data manipulation similarities or differences). Results are summarized by means of the following indicators, on each of the given case studies respectively:

- $MinSim_{fp}$ – the minimum percentage of assessed similarity, per functional process (when compared to all the remaining functional processes in that case study);
- $MinSim_{avg}$ – the average of the minimum values over all the functional processes;
- AvgSim – the average of the assessed similarity over all the functional processes;
- $MaxSim_{fp}$ – the maximum percentage of assessed similarity, per functional process (when compared to all the remaining functional processes in that case study);
- $MaxSim_{avg}$ – the average of the maximum values over all the functional processes.

Table 5. First-order results of functional similarity evaluation.

No.	Case Study ID.	Number of Functional Processes	Avg. Size per Functional Process (in Cfsu)	$MinSim_{fp}$	$MinSim_{avg}$	AvgSim	$MaxSim_{avg}$	$MaxSim_{fp}$
1	ALS	14	4.7	40%	52%	73%	100%	100%
2	SAGA	19	6.2	0%	0%	10%	27%	75%
3	VC	1	9.4	0%	0%	0%	0%	0%
4	HRS	7	12.0	0%	23%	61%	88%	100%
5	LES	15	4.1	0%	0%	8%	37%	67%
6	RC	3	4.0	0%	0%	9%	18%	33%
7	CRS	19	5.1	0%	20%	28%	68%	100%
8	CSPS	27	5.0	0%	0%	9%	45%	75%

It is to be noted that the intent of the proposed approach for the evaluation of functional similarity is not to provide an exact number of candidates for reuse, but a reasonable assessment of that number. Such a reasonable assessment can be very useful to management for planning purposes. For instance, case study 1 (ALS) with 73% for AvgSim and case study 4 (HRS) with 61% for AvgSim have by far the greatest potential for functional reuse. In contrast, case study 5 (LES) with 8% for AvgSim and case study 8 (CSPS) with 9% for AvgSim have very little potential for functional reuse. Should management be interested in investing resources with a view to reaping the benefits of reuse, these are the two case studies that would be good candidates for their investment.

Looking at all the cases in Table 5 except case 3 (VC), there seems to be no specific similarity pattern related to the average size per functional process; that is, systems with similar amounts of data movements per functional process can exhibit very different similarity patterns (see, for instance, cases 5 and 6, or 7 and 8). Case study 3 (VC), with only zero values for the functional similarity indicators, has only a single functional process, and it is trivial that no similarity at all could be found to "other" functional processes (within the same system).

It is to be noted that the comparison process is somewhat time-consuming and theoretically requires comparison of each functional process, out of N, with the remaining N-1 functional processes in the given case study, thus requiring $N^2 - N = N (N - 1)$ comparisons (comparison of a functional process with itself is obviously to be avoided, as it leads to a trivial identity, and is therefore not included in the above numerical assessment results). In practice, however, it is, in fact, quicker. First of all, the comparison is transitive, in the sense that comparing functional process A with functional process B yields the same results (similarity is independent of the comparison direction), thus cutting by half the number of comparisons required. Also, depending on the quality of the measurement records available, some measurement data-filtering can be performed to help accelerate the comparison process; for instance, by filtering on the same type of data movements (E, X, R, W) and/or by filtering only on one particular data group at a time (since data movements over different data groups are not to be considered similar from a functional perspective). A relaxation of the previous criterion could provide some hints for assessing a different kind of similarity (reuse), which we would denote as "technical similarity (reuse)", as opposed to the "functional similarity (reuse)" discussed here.

Finally, we provide a brief note about the many cases showing values for $MinSim_{fp}$ and $MinSim_{avg}$ that both equal 0% (nos. 2, 5, 6, 8, excluding the trivial case 3, in Table 5). This fact represents the simple case where, taking *any* of the functional processes in the system being examined, there is at least one *other* functional process in the system which has nothing in common with the first one taken for comparison. When this is true whatever functional process is being considered first in comparison with the other functional processes, this would mean that the system could be divided into two or more subsystems having potentially nothing in common, from a functional (measurement) point of view; trivially, this would be the case when just one functional process in the system is disjointed from all the remaining functional processes, e.g. a login process which makes use of some "user" data group, which is then not used by any other functional process in the system.

4.2. Second-order evaluation results

By taking into consideration similar data manipulations across the functional processes, a second-order similarity evaluation has been carried out for the eight case studies. Each time two functional processes in the measurement report were identified as having the "same" data movements with the "same" data groups, and the "same" data manipulations regarding the "same" data groups, they were considered to be the 'same' or 'similar'. Similar criteria were considered, as mentioned in section 4.1, when assessing the similarity of data manipulations based on the action-type list proposed in Table 3. In particular, in each case study, data manipulations were taken to be similar only if it could be argued, from the functional requirements, that they involve (i.e. process) the same data group.

Table 6 presents the numerical results of the application of the proposed functional similarity evaluation technique at the second-order stage of the assessment. Results are summarized by means of the same indicators as at the previous stage ($MinSim_{fp}$, $MinSim_{avg}$, AvgSim, $MaxSim_{fp}$, and $MaxSim_{avg}$).

Table 6. Second-order results of functional similarity evaluation.

No.	Case Study I.D.	Number of Functional Processes	Avg. DMan's Per Functional Process (count)	MinSim_{fp}	MinSim_{avg}	AvgSim	MaxSim_{avg}	MaxSim_{fp}
1	ALS	14	1.0	10%	10%	49%	100%	100%
2	SAGA	19	0.9	0%	0%	5%	30%	70%
3	VC	1	5.0	0%	0%	0%	0%	0%
4	HRS	7	2.4	0%	6%	23%	51%	100%
5	LES	15	0.8	0%	0%	4%	28%	70%
6	RC	3	1.3	0%	3%	16%	28%	40%
7	CRS	19	5.1	0%	5%	8%	39%	90%
8	CSPS	27	5.0	0%	0%	3%	20%	70%

With the reminder that the intent of the proposed approach for functional similarity evaluation is not aimed at providing an 'exact number' of candidates for reuse, but rather a reasonable assessment of that number, we can highlight, for instance, that case study 1 (ALS) with 49% for AvgSim and case study 4 (HRS) with 23% for AvgSim still have by far the greatest potential for functional reuse. In contrast, case studies 2 (SAGA), 5 (LES), 7 (CRS), and 8 (CSPS) with 5%, 4%, 8%, and 3%, respectively, for AvgSim have very little potential for functional reuse.

Again, for case study 3 (VC) with only a single functional process, it is trivial that no similarity could be found. It is also worth noting that several functional processes were found, across all the case studies, where no specific data manipulation action was identified according to the proposed list in Table 3. this would be the case, for instance, for a functional process designed to simply "pass over" information between the system and its user by means of data movements, with no optional or specific processing or analysis requirements. Thus, it is possible for a functional process to have no data manipulation.

Comparing the second-order results (Table 6) with the first-order results (Table 5), it should be noted that the average similarity values decrease in most cases (AvgSim for cases 1, 2, 4; from 73% to 49%, from 10% to 5%, and from 62% to 23% respectively). The exact differences in amounts between the cases are affected by the proposed ranges of similarity in the second-order similarity matrix (Table 2). As a trend, adding the data manipulation consideration to the data movement comparison highlights those cases where a strong similarity with respect to sharing the same data movements is not confirmed when considering the actions that are internally performed by the functional processes on the data groups being moved. However, this trend is not a law, as shown for case 6 (RC), where the average similarity increases from 9% to 16%; in this case (a small case study, actually), the similarity with respect to data manipulation enforces (overcomes) the similarity due to the sole data movement comparison. This suggests that maybe the term "approximation order", which we used to label the first- and second-order results, is not accurate: the second-order evaluation adds a *new* similarity aspect to the first-order one, in that it is more than a refinement of the latter.

As was noted for the first-order evaluation, the comparison process proved to be less time-consuming than might be expected, in those cases where the functional processes are grouped by different, non-coupled functional areas (subsystems). While filtering by data group can help in this case, a preliminary (quick) reading of the functional requirements can point the analyst to an optimal way to perform the comparison by avoiding comparing software portions (processes) that do not share data and/or specific purposes. For example, having two separate CRUDs on different data groups (Create, Read, Update, Delete of an Entity "A", and of a different entity "B") would not require any comparison, from a functional perspective (while they would share a maximum technical similarity, as opposed to functional similarity).

More generally, starting from the first-order comparison details (not reported here) highlighted which portions were unlikely be comparable at the second-order stage as well.

The previous comment would also suggest a way to evaluate “technical” similarity, that is, adopting as more general criteria the comparison of data movements and data manipulations, even when different data groups are involved in the functional processes being examined. A special case of “technical” similarity would be, for instance, having totally different data movements (meaning different referenced data groups), while the same data manipulations are performed by the functional processes being compared; high values of such “technical” similarity would likely suggest checking whether or not the two separate functions are really required: it could be more useful to design a single function performing the same manipulation processing over two or more data groups at the same time.

4.3. Sample visualization

A simple suggestion for illustrating pattern similarity across a measured system might be to make use of Data Movement Diagrams (at the first order stage – DMD) or Functional Process Diagrams (at the second order stage – FPD), as introduced in [7]. A sample from [7] is shown in Figure 1 for instance; in this sample, functional processes “a”, “b” and “c”, and functional processes “b”, “c”, and “d”, separately, comprise similar data movements and/or data manipulation types over the object of interest “OoI 1”. This kind of diagram provides an intuitive representation of similarity between the measured functional processes.

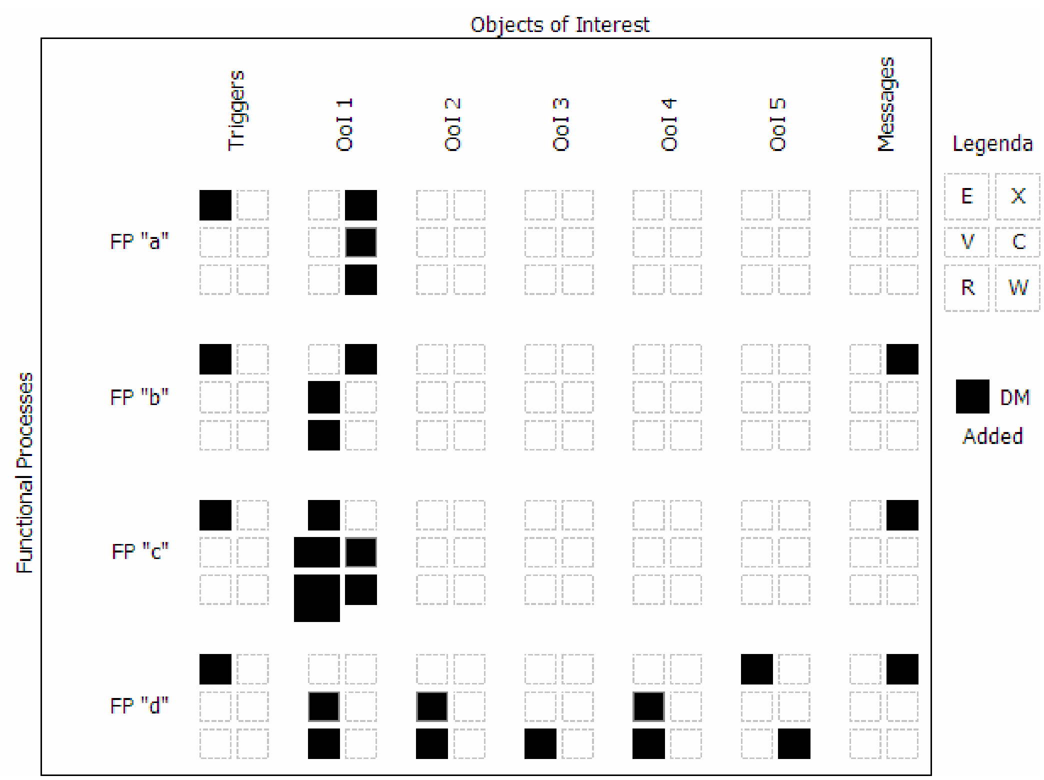

Figure 1. Functional Process Diagram for a generic example. DM = data movement and data manipulation (types); E, X, R, W = Entry, eXit, Read, and Write sub-processes; V, C = validation/check and creation types for data manipulations (see Table 3).

5. Summary and discussion

5.1. Summary

A technique to identify opportunities for reuse based on the similarity between software functions has been proposed and illustrated by means of eight case studies for which detailed functional size measurement results are available. The functional similarity can be assessed at different orders of approximation by comparing data movements only, or data movements and data manipulations, across the functional processes of the system being analyzed. Criteria have been identified to help with the comparison process, and further refinements have been highlighted for future development.

5.2. Discussion

The benefit of the proposed functional reuse evaluation is not to provide precise answers, but to point to the right direction – with a minimum of information, within a fairly short time and with a reasonable degree of confidence that this is indeed the right direction (this 'reasonable degree of confidence' comes from a transparent process with the application of evaluation criteria that are themselves considered as relevant to assess subsets of the overall goal – which is potential reuse). For the actual implementation of reuse, a reuse specialist should then go into the details, working them out one at a time: after analyzing all the candidates for potential functional reuse, he should come up with, for instance, a design that will allow the maximum implementation of reuse. Of course, to arrive at that 'precise' answer will require much more time and effort than the preliminary evaluation of functional similarity proposed here.

There is another consideration with respect to the scope of the proposed approach. Because it is based on records of functional measurement details, this technique is most suitable for the evaluation of functional similarity and possible (functional) reuse. Up to now, there has not been any further investigation into whether or not it could serve as a basis for the evaluation of technical reuse as well, where this term is meant, in general, to refer to software portions handling different data by means of similar algorithms or structures. However, as highlighted when discussing the second-order approximation on the comparison of data movement and data manipulation components, technical reuse should not be excluded a priori.

Finally, the proposed technique requires a set of measurement details as a basis. This requirement is not a true limitation, in the sense that having no measurement details means having no knowledge or control at all of the software being examined.

5.3. Further Developments

This paper was aimed at introducing a functional similarity evaluation technique based on data already collected in measuring functional size using the COSMIC-FFP ISO 19761 standard. The capability of the technique to distinguish between different similarity rankings, thus providing a way to evaluate potential reuse before the real development of a software system, was demonstrated using case studies which included the description of (functional) requirements and the corresponding size measurement results. Further improvements to the proposed technique are possible, by refining the criteria used to compare the measurement elements (data movements and data manipulations across functional processes), by enhancing the action types list for data manipulation comparison (reducing it to ease the comparison process, or extending it to increase similarity evaluation precision). When additional case studies become available to researchers which also include effort details, the relationships between functional size, potential reuse by functional similarity evaluation and effort data will be investigated. Due to the relative ease of application of the comparison criteria, the proposed technique seems promising for real-world application, wherever functional measurement practice is adopted.

6. References

[1] ISO/IEC 19761:2003, "Software engineering – COSMIC-FFP – A functional size measurement method", International Organization for Standardization – ISO, Geneva, 2003.

[2] Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., Symons, C., (2003), "COSMIC-FFP Measurement Manual: The COSMIC Implementation Guide for ISO/IEC 19761: 2003", Version 2.2, January 2003, The Common Software Measurement International Consortium. Online: http://www.gelog.etsmtl.ca/cosmic-ffp/manual.html.

[3] NESMA, "Function Point Analysis for Software Enhancement, Guidelines, v. 1.0", 1998. Online: http://www.nesma.org.

[4] IFPUG, "Function Point Counting Practices Manual", Release 4.2.1, International Function Point Users Group, 2005.

[5] Khelifi A., Abran A., and Buglione L.: A System of References for Software Measurements with ISO 19761 (COSMIC-FFP), 14th International Workshop, IWSM 2004, Berlin, Germany, November 3-5, 2004.

[6] Khelifi, A., "Un referentiel pour la mesure du logiciel avec la norme ISO 19761 – COSMIC-FFP – Une etude exploratoire", PhD thesis, Ecole de technologie superieure, Universite du Quebec, Canada, June 2005.

[7] Santillo, L., " Functional Details Visualization and Classification in the COSMIC FSM Framework", 14th International Workshop, IWSM 2005, Montreal, September 12-14, 2005.