From Function Points to COSMIC - A Transfer Learning Approach for Effort Estimation

Anna Corazza[1], Sergio Di Martino[1], Filomena Ferrucci[2], Carmine Gravino[2], Federica Sarro[3]

[1] Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
Università di Napoli "Federico II"
anna.corazza@unina.it, sergio.dimartino@unina.it
[2] Dipartimento di Informatica
Università di Salerno
fferrucci@unisa.it, gravino@unisa.it
[3] CREST, Department of Computer Science
University College London
f.sarro@ucl.ac.uk

Abstract. Software companies exploit data about completed projects to estimate the development effort required for new projects. Software size is one of the most important information used to this end. However, different methods for sizing software exist and companies may require to migrate to a new method at a certain point. In this case, in order to exploit historical data they need to resize the past projects with the new method. Besides to be expensive, resizing is also often not possible due to the lack of adequate documentation. To support size measurement migration, we propose a transfer learning approach that allows to avoid resizing and is able to estimate the effort of new projects based on the combined use of data about past projects measured with the previous measurement method and projects measured with the new one. To assess our proposal, an empirical analysis is carried out using an industrial dataset of 25 projects. Function Point Analysis and COSMIC are the measurement methods taken into account in the study.

Keywords: Effort estimation; COSMIC; Function Points; Transfer learning

1 Introduction

Software development effort estimation represents a crucial management activity. Software companies exploit data about completed projects to estimate the effort required to develop new projects. Besides the actual effort needed to develop past projects, software size is one of the most important information employed to this end. In this context, Functional Size Measurement (FSM) methods are especially important since they are meant to provide an early software size estimation based on the Functional User Requirements (FURs). Several FSMs exist that differ in several aspects. Function Point Analysis (FPA) [1] was the first FSM method; conceived in the era of transactional systems, it is meant to size a software product by identifying the set of ”features” it provides. COSMIC, initially conceived for real time systems, sizes a software depending on the data movements from/to persistent storage and users, that can be deducted from FURs [2] needed to realize each requirement. COSMIC is considered a 2^{nd} generation FSM method to distinguish it from other previous FSM methods (including FPA) that represent 1^{st} generation FSM. Software companies choose an FSM method based on several criteria, e.g., know-how, customers measurement requirements, organizational policies or also effectiveness. So, the migration from a method to another might be motivated by the changes in one or more of those criteria. Nevertheless, a company that wants to migrate from a measurement method to another and use the new method for effort estimation needs to face the lack of historical data in terms of the new measure. This happens for example to the software companies that would like to migrate from a 1^{st} generation FSM method (e.g., FPA) to a 2^{nd} generation FSM method (COSMIC).

To address the problem, a company could re-measure the past projects with the new measurement method. Besides being expensive, often resizing is also not possible due to the lack of adequate documentation. Another solution could be sizing only the new projects and use public data from other companies (i.e., without-company) until the company builds its own database (i.e., within-company) to predict software development effort. However, there is no evidence in literature that without-company models perform as well as within-company models to predict the effort of new projects [3]. Therefore, the use of statistical conversion equations from FPA to COSMIC has been proposed to automatically re-size past projects (e.g. [4], [5]). Several conversion equations have been proposed, based on the use of different datasets and regression methods. Nevertheless, the effectiveness of this approach strongly depends on the employed conversion equations [6], especially when the conversion equation is obtained from without-company data [7].

In this paper we propose a different approach, based on *transfer learning* and able to estimate the effort of new projects exploiting and adjusting the information gathered about past projects over the time. The approach proposed herein builds adaptive regression models based on the combined use of data on past projects sized with the previous measurement method (source domain) and incoming data about new projects sized with the new measurement method (target domain). In particular, to estimate the effort of new projects sized with COSMIC, we start by applying Least Squares Regression (LSR) to the projects measured with FPA. Then we apply the LSR on the (few) COSMIC points, in combination with a regularization factor that allows the estimator to start from the source initial solution and smoothly adapt to the target domain, until enough information is available in terms of projects measured with COSMIC. In other words, we exploit the knowledge acquired from the source domain as long as we do not have enough points in the target domain.

To assess the effectiveness of the proposal, we carried out an empirical study using an industrial dataset of 25 projects. We use as baseline the predictions obtained with the LSR estimation model based only on FPA sizes. Furthermore, we compare the predictions provided by the proposed approach with respect to those obtained by using only COSMIC sizes. Finally, we also perform a comparison with the predictions obtained by a simple conversion equation.

In the remainder of the paper, Section 2 provides background information on the employed FSM methods and on the conversion equations from FPA to COSMIC. Section 3 introduces the proposed transfer learning approach. Section 4 explains the design of the empirical study and reports its results, while Section 5 discusses the threats to its validity. Related work are presented in Section 6. Section 7 concludes the paper.

2 Functional Size Measures: 1^{st} and 2^{nd} Generation

In the following, we first provide a brief history of FSM methods and then the main notions of FPA and COSMIC methods. We also present the related work on the migration from FPA to COSMIC.

2.1 Functional Measurement Methods

In our investigation we focused on Functional Size Measures, because, differently from dimensional measures, such as Lines of Code, they are particularly suitable to be applied in the early phases of the development lifecycle, when only Functional User Requirements (FURs) are available, being the typical choice for tasks such as estimating a project development effort.

The first FSM method proposed in the literature was FPA, introduced by Albrecht in 1979 [1] as a measure to quantify the functionalities provided by a software from the end-user point of view. Since 1986 FPA is managed by the International Function Point Users Group (IFPUG) [8] and it is named IFPUG FPA (or IFPUG, for short), which has been standardized by ISO as ISO/IEC 20926:2009. FPA has evolved in many different ways (e.g., MkII Function Point, the Boeing 3D Function Points, or the Full Function Point (FFP) [9]). Since these methods are all based on the original formulation by Albrecht, they are also known as 1^{st} generation FSM methods.

At the end of the 90's a group of software measurers formed the *Common Software Measurement International Consortium* (COSMIC) to define a new FSM method to overcome some limitations of the original formulation. The result was the COSMIC-FFP method, which is considered the first "2^{nd} generation FSM method". To highlight this concept, the first version of the method is the 2.0. Important refinements were introduced in 2007 in the version 3.0, named simply COSMIC and standardized as ISO/IEC 19761:2011. The current version of COSMIC is 4.0.1, introduced in April 2015.

In the following we describe the main concepts underlying the IFPUG and COSMIC methods. Among the 1^{st} generation methods, we analyze IFPUG since it is still the most widely used by software practitioners.

2.2 The IFPUG Method

IFPUG sizes an application usually using its FURs. Indeed, to identify the set of "features" provided by the software, each FUR is functionally decomposed into Base Functional Components (BFC), and each BFC is categorized into one of the five *Data* or *Transactional* BFC Types. The Data BFC are defined as follows:

- Internal Logical Files (ILF) are logical, persistent entities maintained by the application to store information of interest.

- External Interface Files (EIF) are logical, persistent entities referenced by the application, but are maintained by another software application.

While the Transactional BFC are defined as follows:

- External Inputs (EI) are logical, elementary business processes crossing the application boundary to maintain the data on an Internal Logical File.
- External Outputs (EO) are logical, elementary business processes that result in data leaving the application boundary to meet a user requirements (e.g., reports, screens).
- External Inquires (EQ) are logical, elementary business processes that consist of a data trigger followed by a retrieval of data that leaves the application boundary (e.g., browsing of data).

Then, the "complexity" of each BFC is assessed through the identification of further attributes (such as the number of data fields to be processed). Once derived this information, a table provided in the IFPUG method [8] specifies the complexity of each function, in terms of Unadjusted Function Points (UFP). The sum of all these UFPs gives the functional size of the application. Subsequently, a Value Adjustment Factor (VAF) can be computed to take into account some non-functional requirements, such as Performances, Reusability, and so on. The final size of the application in terms of Function Points is given by $FP = UFP \cdot VAF$. For more details on the IFPUG method, readers may refer to the counting manual [8].

2.3 The COSMIC Method

The basic idea underlying the COSMIC method is that, for many types of software, most of the development effort is devoted to handle data movements from/to persistent storage and users. Thus, the number of these data movements can provide a meaningful sight of the system size [2]. To identify and count these data movements, the measurement process consists of three phases [2]:

1. The *Measurement Strategy* phase is meant to define, among others, the *purpose* of the measurement, the *scope* (i.e. the set of FUR to be included in the measurement), and the *functional users* of each piece of software.
2. The *Mapping Phase* requires to express each FUR in the form required by the *COSMIC Generic Software Model*. This model, necessary to identify the key elements to be measured, assumes that (i) each FUR can be mapped into a unique *functional process*, meant as a cohesive and independently executable set of data movements, (ii) each functional process consists of *sub-processes*, and (iii) each sub-process may be either a *data movement* or a data manipulation.
   As depicted in Figure 1, data movements are defined as follows:
   - An Entry (E) moves a data group from a functional user across the boundary into the functional process where it is required.
   - An Exit (X) moves a data group from a functional process across the boundary to the functional user that requires it.
   - A Read (R) moves a data group from persistent storage within each of the functional process that requires it.

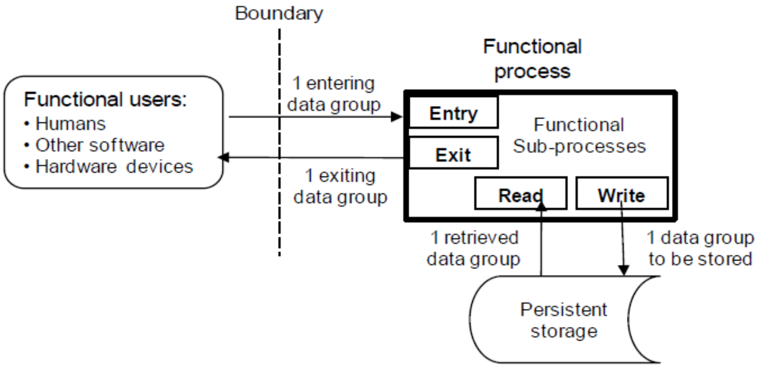

Fig. 1: The four types of COSMIC Data Movements, and their relationship with a Functional Process [2]

- A Write (W) moves a data group lying inside a functional process to persistent storage.

3. The *Measurement Phase*, where the data movements of each functional process have to be identified and counted. Each of them is counted as 1 COSMIC Function Point (CFP) that is the COSMIC measurement unit. Thus, the size of an application within a defined scope is obtained by summing the sizes of all the functional processes within the scope.

For more details about the COSMIC method, readers are referred to the COSMIC Measurement Manual [2].

2.4 Converting Function Points into COSMIC

From the brief descriptions of the two FSM methods reported in the previous sections, we can see that FPA and COSMIC consider different aspects of a software system for its size measurement, since they are based on different basic functional components [10]. Thus, "exact conversion formulae from sizes measured with a 1^{st} generation method to COSMIC sizes are impossible" [11].

A possible way to address the problem, also suggested in the COSMIC documentation [11], is to search for some "statistically-based conversion formulae". Some researchers have been investigating the suitability and the effectiveness of such an approach by building conversion equations for different data sets. In particular, linear and non-linear equations have been built by applying the linear regression analysis on the raw data and on the log-transformed data, respectively [4]. Also, more sophisticated techniques, such as piecewise regression, have been used to build non-linear models [12].

The results reported in the literature [4] [12] [13] [14] [15] [16] [17] [18] [19] [20] reveal that a statistical conversion is possible, thus supporting the suggestions provided in the COSMIC documentation [11]. The studies also showed that both linear and non-linear models should be analyzed to identify the best correlation. Furthermore, more complex techniques, such as piecewise regression [12], did not provide significantly better results, being at the same time hardly applicable.

As results of these investigations different conversion equations have been proposed, that might be exploited to convert historical FP based data sets into COSMIC based data sets. Among them, empirical evidence seem to suggest that a trivial 1 CFP $\cong$ 1 FP conversion could be applied to have a quick and dirty approximation of the size in terms of COSMIC [4], even if authors pointed out that "'1 to 1 conversion cannot be attributed to anything other than an influential coincidence"' [4], as FPA and COSMIC are meant to measure different attributes of the software.

In [6] we analyzed the effectiveness of all the conversion equations proposed in the literature for effort estimation purposes. The obtained results revealed that the effectiveness depends on the employed conversion equations. No guidelines can be provided to the software company on how to carry out the selection. Furthermore, the use of without-company conversion equations resulted to be worse than within-company conversion equations [7].

For this reason, we decided to investigate a different strategy based on the idea of *transfer learning* that has been successfully applied in other contexts [21].

3 The Proposed Trasfer Learning Approach

To support a company in the migration from a size measure to another, we would like to find a solution aiming at transferring the knowledge about the relationship between software size and development effort, extracted from a *Source Domain* (SD), where each past project is sized in terms of Function Points, to a *Target Domain* (TD), where the size measure is COSMIC. Let us note that, given this problem definition, the SD and the TD have different feature space and distribution. As a consequence, the most of the traditional machine learning methods cannot be applied [21].

Since this is a kind of problem arising in many scenarios, also outside software engineering, the research community has provided a new family of approaches, known as *transfer learning* [21]. Indeed, transfer learning is a general framework including several techniques to bring some knowledge from an SD into a TD on a given task that could be classification, regression or clustering. More formally, given an SD with a *Source Task* (ST) and a TD with a *Target Task* (TT), a transfer learner is aimed at improving the effectiveness of the prediction function in the TD using the knowledge of SD and ST [21].

Given the combinations of differences among Domains and among Tasks, there is a taxonomy of transfer learning approaches, as extensively discussed in [21]. In our case, like the most of transfer learning problems in software engineering [22], we are in the so-called *transductive transfer learning* scenario, where ST and TT are the same (to build an effort estimation model), while SD and TD are different [23]. Moreover, in our case, also the feature spaces between the SD and TD are different, since based on different size measures (Function Points in the SD and COSMIC in the TD).

The goal of our proposal is to transfer the knowledge from an SD based on FP to a TS based on COSMIC, to build an effort estimation model in the TD. This model will be built incrementally, using any new project developed by the company during the migration, and sized only with the new measure. More in details, as we want to learn from both the domains SD and TD, we consider a training set composed by two parts, the source training set T_{FP}, whose points are expressed in terms of FP, and the target training set T_{CFP}, in terms of CFP. The former is composed by $m_{\text{FP}} = |T_{\text{FP}}|$ points represented by a feature vector x_{FP}, the latter by $m_{\text{CFP}} = |T_{\text{CFP}}|$ points in the target domain, corresponding to the feature vector x_{CFP}. Clearly, since FPA and COSMIC are able to express the size of a software with one number, the dimension of feature vectors in both domains is equal to 1. Furthermore, in both domains a real number y is associated to each item, representing the actual development effort, expressed in person/hours. In conclusion, we adopt the following notation: $T_{\text{FP}} = \{(x^i_{\text{FP}}, y^i), 1 \leq i \leq m_{\text{FP}}\}$ and $T_{\text{CFP}} = \{(x^i_{\text{CFP}}, y^i), 1 \leq i \leq m_{\text{CFP}}\}$.

In the proposed approach we use as estimation technique the Least Square Regression (LSR), since it is a simple but effective technique widely and successfully employed in the industrial context and in several researches to estimate development effort (see e.g., [24] [25] [26]). If we had enough previous projects measured with COSMIC, we could simply disregard the dataset based on Function Point and apply LSR to construct an effort estimation equation directly in the target domain. In this case, the LSR equation can be written as follows:

$$(a^*_{\text{CFP}}, b^*_{\text{CFP}}) = \arg\min_{a,b} \sum_{x^i_{\text{CFP}} \in T_{\text{CFP}}} \left(a x^i_{\text{CFP}} + b - y^i\right)^2 \tag{1}$$

where a^*_{CFP} and b^*_{CFP} represent the coefficient and the intercept of the linear equation minimizing the sum of the squares of the errors.

However, since the proposed approach is meant to support a company at the beginning of the migration when the number of examples in T_{CFP} (i.e. completed projects whose size is measured in CFP) is not sufficient for an effective learning of the relationship between software size and development effort, we extract as much information as possible from the source domain to improve regression in the target domain.

To this aim, we apply LSR to estimate the solution in the source domain:

$$(a^*_{\text{FP}}, b^*_{\text{FP}}) = \arg\min_{a,b} \sum_{x^i_{\text{FP}} \in T_{\text{FP}}} \left(a x^i_{\text{FP}} + b - y^i\right)^2 . \tag{2}$$

Parameters $(a^*_{\text{FP}}, b^*_{\text{FP}})$ represent the information that we extract from the SD and will inject in the final estimator.

Now, starting from the observation that a relationship 1 to 1 between FP and COMSIC could be a basic approximation [4], we want that this estimator considers these parameters as a good approximation until enough information is available in terms of projects measured with COSMIC. Better than that, we want that, starting from this initial estimation, it smoothly adapts to the new COSMIC-based domain.

Regularization factors are often used in machine learning to minimize the value of parameters. In our case, however, the introduction of such a factor aims to pushing the parameters of the LSR models we are trying to learn in the TD to be as similar as possible to the ones trained in the SD. A similar idea is considered in [27] to generalize an approach proposed by [28] for maximum entropy classification. In that case, the approach is considered among the baselines and more complex approaches outperform it. However, in the natural language processing field both the number of features and the source domain training set are much larger than in the case of effort estimation. On the other hand, a crucial aspect of the effort estimation domain is the usual scarcity of past information: in fact usually a software company has just a limited number of both points (past projects) and features (software size) with respect to most machine learning problems, and such approaches would risk overfitting. Therefore, we need to find a good compromise between the effectiveness of the approach and the risk of overfitting.

Thus, we propose a modification of the LSR equation in the TD by introducing a regularization factor that has the effect of favoring the solution which is as similar as possible to the parameters $(a^*_{\text{FP}}, b^*_{\text{FP}})$ found in the SD:

$$(a^*_C, b^*_C) = \arg\min_{a,b} \left((1-\lambda) \sum_{x^i_{\text{CFP}} \in T_{\text{CFP}}} \left(a x^i_{\text{CFP}} + b - y^i \right)^2 + \lambda \left((a - a^*_{\text{FP}})^2 + (b - b^*_{\text{FP}})^2 \right) \right). \quad (3)$$

The weight of the regularization factor is controlled by the value of λ: when its value is large, the resulting parameters will be more similar to the optimal SD solution, due to the effect of the regularization factor. On the contrary, when its value is lower, the resulting parameters will depend more on the new projects in the target training set. At the limit, for a null value of λ, the estimation will be only based on the projects measured with COSMIC, which is the ideal situation when enough observations with the new measures have been collected. After some experiments, we found that a simple way to define this factor is $\lambda = \frac{1}{m_{\text{CFP}} + \epsilon}$, where ϵ is a small number which avoids that λ goes to infinity in the initial situation, that is when m_{CFP} approaches to zero. The rationale behind this definition of λ is that the more CFP points we get, the less is the importance of the knowledge extracted from the FP source domain.

4 Empirical Study

In this section we present the design and the results of the empirical study we performed to assess the effectiveness of our transfer learning based approach for effort estimation. To this aim, we defined the following research question:

RQ : Is the proposed transfer learning based approach good for effort estimation when migrating from Function Points to COSMIC?

4.1 Data Set

The data set considered in our study includes information about 25 Web applications developed by an Italian medium-sized software company, whose core business is the development of enterprise information systems, mainly for local and central government. In particular, the set of Web applications includes e-government, e-banking, Web portals, and Intranet applications. All the projects were developed with SUN J2EE or

Table 1: Descriptive statistics of EFF, CFP and FP

Var	Obs	Min	Max	Mean	Median	Std Dev
EFF	25	782	4537	2577.00	2686	988.14
CFP	25	163	1090	602.04	611	268.47
FP	25	89	915	366.76	304	208.65

Microsoft .NET technologies. Oracle has been the most commonly adopted DBMS, but also SQL Server, Access and MySQL were employed in some of these projects.

As for the collection of the information, the software company used timesheets to keep track of the Web application development effort. In particular, each team member annotated the information about his/her development effort on each project every day, and weekly each project manager stored the sum of the efforts for the team. Furthermore, to collect all the significant information to calculate the values of the size measure in terms of COSMIC, we defined a template to be filled in by the project managers. All the project managers were trained on the use of the questionnaires. One of the authors analyzed the filled templates and the analysis and design documents, in order to cross-check the provided information. The same author calculated the values of the size measure. As for the calculation of the size in terms of IFPUG, the company has always applied this FSM method to measure its past applications.

Table 1 reports on some summary statistics related to the 25 Web applications employed in our study[4]. The variables are EFF, i.e., the actual effort expressed in terms of person-hours, CFP, expressed in terms of number of COSMIC Function Points, and FP, expressed in terms of number of Function Points.

4.2 Validation Method

To assess the prediction models we performed a cross validation, by considering a training set made of N points for which we have both Function Points and COSMIC measures (i.e., FP and CFP variables). We wanted to evaluate the performance of the proposed approach as a function of the dimension of the source training set m_{FP}: that is, we took the FP for m_{FP} points, and the CFP for the remaining training points, which formed the target training set.

Although the dimension of our data set is reasonable for the task, it is quite small for evaluation. In order to exploit it as much as possible, for each value of m_{FP}, we adopted the Leave-One-Out (LOO) protocol: at each iteration we kept a point for testing, while we split the remaining $N-1$ points in source and target training sets, built the estimator and applied it to the test point. All in all, as the dimension of the source training set is m_{FP}, the target training set had size $m_{\text{CFP}} = N - m_{\text{FP}} - 1$. The performance on the single test points was then merged to obtain the performance on the complete data set.

In the proposed solution, the sequence in which projects are considered (being in m_{FP} or in m_{CFP}) may strongly impact the results. In order to avoid the chance of obtaining an extremely favorable or unfavorable disposition of points between the two training

[4] Raw data cannot be revealed because of a Non Disclosure Agreement with the software company.

sets, we randomized the experiment by repeating the LOO procedure on 100 random permutations of the data set for each value of m_{FP}, and then considering the average of the performance. Furthermore, on the basis of the standard deviation of performance, we can also estimate its confidence interval.

4.3 Employed Benchmarks

Since we wanted to verify whether the proposed transfer learning approach can support the companies in the migration from FPA to COSMIC for development effort estimation, as baseline we considered the predictions obtained with the estimation model obtained from the Function Points sizes (named *FP model* in the following). The rationale is that the company should achieve effort predictions that are at least not significantly worse than those it would obtain going on with FPA. Furthermore, we compared the predictions obtained by the model built with the transfer learning based approach (named *CFP$_{TL}$ model* in the following) with those obtained by exploiting the estimation model based on the measured COSMIC sizes (named *CFP model* in the following). Indeed, this represent the accuracy the company could obtain by a dataset whose points are all measured in COSMIC. Finally, we consider also the 1 CFP $\cong$ 1 FP conversion, since it is the starting point of our transfer learner. In the following we denote by *CFP$_{FP}$* this model. It is important to note that the FP and CFP$_{\mathrm{FP}}$ models provide different size predictions because in the application of the LOO for the former we measured the observations in the training and test sets in terms of FP while for the latter the observation in the test set is measured in terms of CFP.

The three estimation models for the three baselines were built using LSR employing FP, CFP, and CFP$_{\mathrm{FP}}$ as independent variables, respectively. The dependent variable was EFF for all the three models.

4.4 Evaluation Criteria

The accuracy of the obtained prediction was evaluated exploiting Absolute Residuals (AR), i.e., |*Actual* - *Predicted*|, where *Actual* and *Predicted* are the actual and the estimated efforts, respectively. To have a summary measure for comparing the different estimation approaches we employed Mean of AR (MAR)[29]. In particular the results are presented through a graphical representation, namely a simple plot. A method X is better than another method Y if the MAR value obtained with X is less than the one of obtained with Y.

Moreover, we tested whether there was a statistically significant difference between the absolute residuals achieved with the CFP$_{\mathrm{TL}}$ model and those obtained with the FP, CFP, and CFP$_{\mathrm{FP}}$ models. The results were intended as statistically significant at $\alpha = 0.05$. In order to have also an indication of the practical/managerial significance of the results, we verified the effect size, which is a simple way of quantifying the standardized difference between two groups. In particular, we employed the Cliffs d non-parametric effect size measure because it is suitable to compute the magnitude of the difference when a non parametric test is used. In the empirical software engineering field, the magnitude of the effect sizes measured using the Cliffs d can be classified

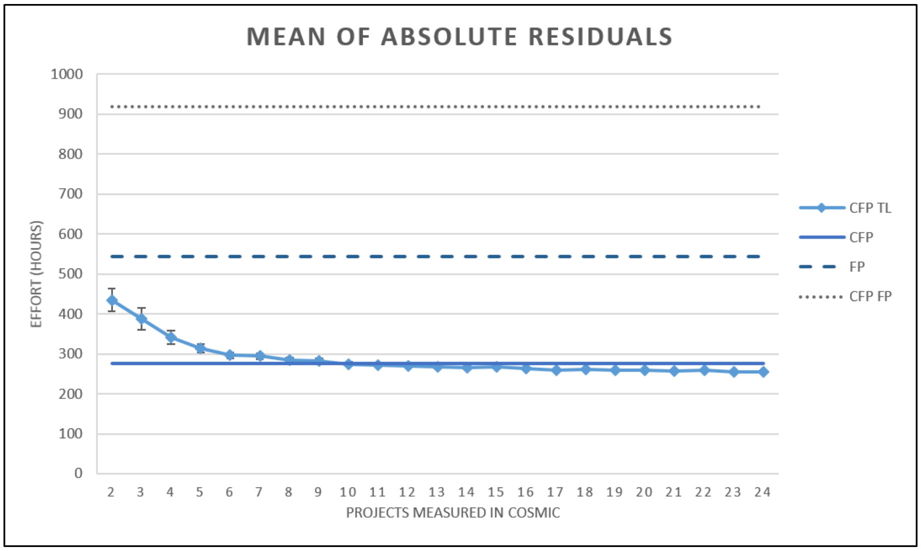

Fig. 2: Results of the study in terms of MAR

as follows: negligible (d <0.147), small (0.147 to 0.33), medium (0.33 to 0.474), and large (d >0.474) [30].

4.5 Results

Figure 2 shows the main results we obtained in the study. In particular, we have reported the MAR values plus the standard deviation we got over 100 random sequences of projects using the CFP_{TL} estimation model. This is done on a range from 2 to 24 projects measured with COSMIC. This because we cannot perform LSR on less than 2 points. The figure also shows the MAR values we obtained using the FP, CFP_{FP}, and CFP models. For these three models, we exploited the leave-one-out cross validation on the entire data set of 25 projects to obtain the effort predictions and the corresponding mean of the absolute residuals.

We can observe that the MAR values achieved with the CFP_{TL} estimation model are lower than those obtained with the FP and CFP_{FP} models. Thus, the effort predictions obtained with the proposed transfer learning based approach are better than those obtained with the model based on Function Points sizes and those achieved using the estimated COSMIC sizes (i.e., those considering the assumption 1 CFP $\cong$ 1 FP). In particular, the MAR value achieved with the CFP_{FP} model is about three times higher than the one obtained with the CFP_{TL} model, thus highlighting much better results with the transfer learning approach. The values of MAR achieved with the CFP_{TL} model are about two times lower than the MAR value obtained with the FP model.

The results achieved in terms of MAR are confirmed by the performed statistical tests. Indeed, the performed Mann-Whitney test revealed that the absolute residuals obtained with the CFP_{TL} models are significantly lower than those obtained with the FP model (p-value = 0.008) with a medium effect size (d=0.443). Similarly, the predictions obtained with the CFP_{TL} model are significantly better than those obtained with the CFP_{FP} model (p-value < 0.001) with a large effect size (d=0.917).

These results clearly reveal that the company involved in our study can abandon the FP based model and employ the CFP_{TL} model for effort estimation during the migration (from Function Points) to COSMIC as method for sizing their applications since their second project in COSMIC.

To further highlight the potential of the proposed transfer learning based approach we have also compared the effort predictions obtained with the CFP_{TL} model with those achieved using the CFP model. The results reported in Figure 2 show that after measuring 8-9 Web applications with COSMIC and using the obtained sizes in the proposed transfer learning based approach the obtained effort predictions are very close to the ones achieved using the model based only on the measured COSMIC sizes.

The results in Figure 2 also show that for several points (i.e., from 13 to 24) the effort predictions obtained with CFP_{TL} model are even slightly better that those achieved with the CFP model, since the corresponding MAR values are lower than those achieved with CFP model. This is a rather surpising results that can be justified as follows. While usually the Absolute Residuals with COSMIC are by far lower than those with FPA, on one specific project the Absolute Residuals with COSMIC are much higher, being almost double of FPA. This of course has a strong impact on the cumulative measure MAR. Probably, when using the transfer learner equation, this problem is mitigated by the 100 runs. However, the difference achieved in the predictions is not statistically significant (p-value = 0.122) with a small effect size (d=0.267). We think that this point deserves further investigation in the future.

The results presented and discussed above allow us to positively answer our research question: *the proposed transfer learning based approach is good for effort estimation when migrating from Function Points to COSMIC*.

5 Threats to Validity

It is widely recognized that several factors can bias the construct, internal, external, and conclusion validity of empirical studies [31].

As for the construct validity, how to collect information to determine size measures and actual effort represents a crucial aspect [32]. As described in Section 4, we have supervised the procedure employed by the involved software company to carefully collect the information we needed for the empirical analysis. In particular, we tried to perform the data collection task in a controlled and uniform fashion. Of course we have to take into account that empirical studies do not ensure the level of confidence achieved with controlled experiments.

Some factors should be taken into account for the internal validity: subjects' authoring and reliability of the data and lack of standardization [31, 33, 34]. The managers involved in the study were professionals who worked in the software company. No initial selection of the subjects was carried out, so no bias has been apparently introduced. Moreover, the software applications were developed with technologies and methodologies that subjects had experienced. Consequently, confounding effects from the employed methods and tools should be excluded. As for the reliability of the data and lack of standardization, the used questionnaires were the same for all the Web applications, and the project managers were instructed on how to fill them in, to correctly provide the required information. Instrumentation effects in general did not occur in this kind of studies.

As for the conclusion validity, we carefully applied the estimation methods and the statistical tests, verifying all the required assumptions (e.g., the hypotheses underlying the application of linear regression analysis).

With regard to the external validity, we are confident that the type of the analyzed Web applications did not bias the validity of the achieved results, since for their functionalities, target platforms, and complexity they can be considered representative samples of typical current Web applications. Another threat could be the fact that we exploited only applications from one company. To the best of our knowledge, there is only one data set that contains (Web and non-Web) applications from different company, i.e., ISBSG. However, in our analysis we were interested in analyzing the experience and the possibilities for the migration among size measures for a single company developing Web applications. Nevertheless, it is recognized that the results obtained for a given company might not hold for others. Indeed, each development context might be characterized by some specific project and human factors, such as development process, developer experience, application domain, tools, technologies used, time, and budget constraints that could influence the results [35].

6 Related Work on Transfer Learning in Software Engineering

Transfer learning techniques have been already applied in software engineering in the last years, showing their potential. Some studies applied them in the field of defect prediction. Among them, Zimmermann et al. [36] found that defect predictors performed worse when trained on cross-application data than from within-application data. Other recent studies on the use of TL for defect predictions are those by Ma et al. [37] and Nam et al. [38].

Focusing in the field of effort estimation, some studies have been done in the past to migrate estimation models among companies. A survey can be found in [3, 24]. More recently, transfer learning approaches have also been proposed for effort estimation [22, 39–41], as a suitable way to integrate data from different companies and different time frames (which is a different problem from the one we are willing to address). To the best of our knowledge, no one has ever proposed a migration strategy from a size measure to another, using transfer learning approaches.

7 Conclusion

We have investigated the problem of migrating from a measurement method to another and use the new size measure for effort estimation purposes. The subject of our study was a company that decided to migrate from FPA [1] to COSMIC [2], which represents one of the most recent and common transitions we are observing in the context of software measurement. The problem for a company mainly consists in the lack of enough data (in terms of the new measurement method) to build an estimation model. A simple way to overcome this problem could be re-sizing the past projects with the new measurement method. However, besides to be expensive, resizing is also not always possible due to the lack of adequate documentation. As alternative, in the paper we propose to exploit a transfer learning approach able to estimate the effort of new projects exploiting and adjusting the information gathered about past projects over the time. In particular, we aimed at transferring the knowledge extracted from a source domain, in this case represented by projects for which we have the Function Points metrics, to the target domain, where points are represented by COSMIC. As estimation technique we applied the LSR, adapted by using a regularization factor that allows the estimator to start from an initial estimation in terms of Function Points sizes and smoothly adapt to the new COSMIC domain, until enough information is available in terms of projects measured with COSMIC.

To assess the proposed transfer learning approach, we have performed an empirical study using an industrial dataset of 25 projects and employing leave-one-out cross validation as validation strategy. The results have revealed that the effort estimations obtained with the proposed transfer learning approach are significant better than those achieved with a Function Points based estimation model. Furthermore, the predictions achieved with the proposed approach are quite close with those achieved by employing an estimation model exploiting only COSMIC sizes. Thus, the proposed transfer learning based approach is good for effort estimation and the company involved in our study can employ it for effort estimation purposes during for migration from a 1^{st} generation FSM method (i.e., Function Points Analysis) to a 2^{nd} generation FSM method (i.e., COSMIC).

Concerning future work, several directions could be consider for our research. First of all, we intend to replicate the study with further datasets, also considering different types of software projects and a larger number of points. Moreover, we intend to verify whether conversion equations built on external datasets could be employed in the application of the method to perform the adaptation towards the COSMIC domain more rapidly. Besides the COSMIC and Functions Points based measurement methods, the migration problem could regard other size measurement approaches, e.g., extension of Function Points [9]. So, in the future we could consider the applicability of the proposed transfer learning approaches in other measurement contexts.

References

1. Albrecht, A.: Measuring Application Development Productivity. In: Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium. (1979) 83–92
2. Abran, A., Desharnais, J., Lesterhuis, A., Londeix, B., Meli, R., Morris, P., Oligny, S., ONeil, M., Rollo, T., Rule, G., Santillo, L., Symons, C., Toivonen, H.: The COSMIC Functional Size Measurement Method Measurement Manual, version 4.0.1 (2015)
3. Mendes, E., Kalinowski, M., Martins, D., Ferrucci, F., Sarro, F.: Cross- vs. within-company cost estimation studies revisited: An extended systematic review. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. EASE'14 (2014) 12:1–12:10

4. Cuadrado-Gallego, J.J., Buglione, L., Domínguez-Alda, M.J., Sevilla, M.F.d., Antonio Gutierrez de Mesa, J., Demirors, O.: An experimental study on the conversion between IFPUG and COSMIC functional size measurement units. Information & Software Technology **52**(3) (2010) 347–357
5. Lavazza, L.: An evaluation of the statistical convertibility of function points into cosmic function points. Empirical Software Engineering **19**(4) (August 2014) 1075–1110
6. Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F.: Web Effort Estimation: Function Points Analysis vs COSMIC. submitted to an International Journal for review
7. Ferrucci, F., Gravino, C., Sarro, F.: Conversion from IFPUG FPA to COSMIC: within-vs without-company equations. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, 2014. (2014) 293–300
8. IFPUG: International Function Point Users Group - www.ifpug.org
9. Çigdem Gencel, Demirörs, O.: Functional size measurement revisited. ACM Transactions on Software Engineering Methodology. **17**(3) (2008)
10. Van Heeringen, H.: Changing from FPA to COSMIC. A transition framework. In: Software Measurement European Forum. (2007)
11. Abran, A., B.Londeix, O'Neill, M., Santillo, L., Vogelezang, F., Desharnais, J.M., Morris, P., Rollo, T., Symons, C., Lesterhuis, A., Oligny, S., Rule, G., Toivonen, H.: The COSMIC Functional Size Measurement Method, Version 3.0, Advanced and Related Topics (2007)
12. Lavazza, L., Morasca, S.: Convertibility of Function Points into COSMIC Function Points: A study using Piecewise Linear Regression. Information & Software Technology **53**(8) (2011) 874–884
13. Ferrucci, F., Gravino, C., Sarro, F.: A case study on the conversion of function points into cosmic. In: In the Proceedings of the 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2011. (2011) 461–464
14. Abran, A. In: Convertibility across Measurement Methods. John Wiley & Sons, Inc. (2010) 269–280
15. Ho, V., Abran, A., Fetcke, T.: A Comparative Study Case of COSMIC, Full Function Point and IFPUG Methods. Technical report, Département dinformatique, Université du Quebec á Montréal, Canada, (1999)
16. Desharnais, J., Abran, A., Cuadrado-Gallego, J.: Convertibility of function points to COSMIC: identification and analysis of functional outliers. In: Proceedings of the International Workshop on Software Measurement, Shaker-Verlag (2007) 130146
17. Abran, A., Desharnais, J., Azziz, F.: Measurement convertibility: from function points to COSMIC. In: Proceedings of the International Workshop on Software Measurement, Shaker-Verlag (2005) 227240
18. Abualkishik, A.Z., Desharnais, J.M., Khelifi, A., Ghani, A.A.A., Atan, R.B., Selamat, M.H.: An exploratory study on the accuracy of FPA to COSMIC measurement method conversion types. Information & Software Technology **54**(11) (2012) 1250–1264
19. Çigdem Gencel, Bideau, C.: Exploring the Convertibility between IFPUG and COSMIC Function Points: Preliminary Findings. In: Proceedings of International Conference on Software Process and Product Measurement. (2012) 170–177
20. Lavazza, L., Bianco, V.D., Liu, G.: Analytical convertibility of functional size measures: A tool-based approach. In: Proceedings of International Conference on Software Process and Product Measurement. (2012) 160–169
21. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10) (2010) 1345–1359
22. Kocaguneli, E., Menzies, T., Mendes, E.: Transfer learning in effort estimation. Empirical Softw. Engg. **20**(3) (June 2015) 813–843

23. Arnold, A., Nallapati, R., Cohen, W.W.: A comparative study of methods for transductive transfer learning. In: Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, IEEE (2007) 77–82
24. Kitchenham, B., Mendes, E., Travassos, G.: Cross versus Within-Company Cost Estimation Studies: A systematic Review. IEEE Transaction on Software Engineering **33**(5) (2007) 316–329
25. Mendes, E., Di Martino, S., Ferrucci, F., Gravino, C.: Effort estimation: how valuable is it for a Web company to use a cross-company data set, compared to using its own single-company data set? In: Proceedings of the 6th International World Wide Web Conference, ACM press (2007) 83–93
26. Menzies, T., Chen, Z., Hihn, J., Lum, K.: Selecting Best Practices for Effort Estimation. IEEE Transaction on Software Engineering **32**(11) (2006) 883–895
27. Daumé, H.: Frustratingly Easy Domain Adaptation. In: Proc. of ACL 2007. (2007)
28. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: Little data can help a lot. In Lin, D., Wu, D., eds.: Proceedings of EMNLP 2004, Barcelona, Spain, Association for Computational Linguistics (July 2004) 285–292
29. Shepperd, M.J., MacDonell, S.G.: Evaluating prediction systems in software project estimation. Information & Software Technology **54**(8) (2012) 820–827
30. Kampenes, V., Dyba, T., Hannay, J., Sjoberg, I.: A systematic review of effect size in software engineering experiments. Information and Software Technology **4**(11-12) (2007) 1073–1086
31. Mendes, E., Counsell, S., Mosley, N., Triggs, C., Watson, I.: A Comparative Study of Cost Estimation Models for Web Hypermedia Applications. Empirical Software Engineering **8**(23) (2003) 163–196
32. Kaner, C., Bond, W.: Software Engineering Metrics: What Do They Measure and How Do We Know? In: Proceedings of the International Software Metrics Symposium, IEEE press (2004)
33. Mendes, E., Counsell, S., Mosley, N.: Comparison of Web Size Measures for Predicting Web Design and Authoring Effort. IEE Proceedings-Software **149**(3) (2002) 86–92
34. Kitchenham, B., Pickard, L., MacDonell, S., Shepperd, M.: What accuracy statistics really measure. IEE Proceedings Software **148**(3) (2001) 81–85
35. Briand, L.C., Wüst, J.: Modeling Development Effort in Object-Oriented Systems Using Design Properties. IEEE Transaction on Software Engineering **27**(11) (2001) 963–986
36. Zimmermann, T., Nagappan, N., Gall, H., Giger, E., Murphy, B.: Cross-project defect prediction: a large scale experiment on data vs. domain vs. process. In: Proceedings of the 7th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2009. (2009) 91–100
37. Ma, Y., Luo, G., Zeng, X., Chen, A.: Transfer learning for cross-company software defect prediction. Information and Software Technology **54**(3) (2012) 248–256
38. Nam, J., Pan, S.J., Kim, S.: Transfer defect learning. In: Proceedings of the 2013 International Conference on Software Engineering, IEEE Press (2013) 382–391
39. Ferrucci, F., Mendes, E., Sarro, F.: Web effort estimation: The value of cross-company data set compared to single-company data set. In: Proceedings of the 8th International Conference on Predictive Models in Software Engineering. PROMISE ’12 (2012) 29–38
40. Minku, L.L., Yao, X.: How to make best use of cross-company data in software effort estimation? In: Proceedings of the 36th International Conference on Software Engineering. ICSE 2014 (2014) 446–456
41. Minku, L.L., Sarro, F., Mendes, E., Ferrucci, F.: How to make best use of cross-company data for web effort estimation? In: Proceedings of the 9th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM 2015 (2015)