

A COSMIC-FFP Based Method to Estimate Web Application Development Effort

Gennaro Costagliola, Filomena Ferrucci, Carmine Gravino, Genoveffa Tortora, and
Giuliana Vitiello

Dipartimento di Matematica e Informatica, Università degli Studi di Salerno
Via Ponte Don Melillo, 84084 Fisciano (SA), Italy
{gcostagliola, fferrucci, gravino, tortora, gvitiello}@unisa.it

Abstract. In the paper we address the problem of estimating the effort required to develop dynamic web applications. In particular, we provide an adaptation of the *Cosmic Full Function Point* method to be applied on design documents for counting data movements. We also describe the empirical analysis carried out to verify the usefulness of the method for predicting web application development effort.

1 Introduction

In the present paper we address the problem of estimating the effort required to develop web applications. This represents an emerging issue in the field of web engineering, due to the dramatic increasing of complexity and size of web applications, and the consequent demand for tools supporting project development planning with reliable cost and effort estimations [2,6,7,8,9,10,11].

In [9] Rollo employed *COSMIC-FFP* (*Cosmic Full Function Point*) [5], an adaptation of *Function Points* [1] originally defined for real-time applications, to measure the functional size of an Internet bank system. Following his suggestion in [6] Mendes *et al.* provided a formal method which adopts *COSMIC-FFP* to measure size of static hypermedia web applications. In the paper we propose to apply the *COSMIC-FFP* method also to dynamic applications. Indeed, since *COSMIC-FFP* measure is focused on the counting of data movements, it turns out to be suitable for client-server applications, which are characterized by large amounts of data movements. To provide an early size estimation, we propose to apply the method on design documents. In particular, we extend the proposal by Mendes *et al.*, by defining some rules that allow us to measure functional size of client-server applications, using class diagrams. Moreover, we also report on an initial empirical validation of the approach performed on a set of dynamic applications developed by undergraduate students of two academic courses on web engineering.

The paper is organized as follows. In Section 2 we describe the rules to adapt *COSMIC-FFP* counting to class diagrams describing dynamic web applications. Section 3 presents the results of the empirical analysis carried out so far. Section 4 concludes the paper giving some final remarks and discussion on future work.

2 Applying COSMIC-FFP to Dynamic Hypermedia Applications

The *COSMIC-FFP* method is focused on counting data movements from *Functional User Requirements* to obtain the functional size of the software, expressed in terms of *CFSU* (*Cosmic Functional Size Unit*) [5]. Each *CFSU* corresponds to either an *entry*, or an *exit*, or a *read*, or a *write*.

In the sequel, suitable rules are given for counting data movements of dynamic web applications using class diagrams, expressed in terms of the UML notation for the web proposed in [3].

1. For each static web page count $1 \text{ entry} + 1 \text{ read} + 1 \text{ exit}$. Indeed, an entry is sent to the application by requesting the client page (*entry*), the page is read from the web server (*read*) and then shown to the user (*exit*).
2. For each multimedia component, which is visualized after an explicit request of the client, count $C \cdot (1 \text{ entry} + 1 \text{ read} + 1 \text{ exit})$. C denotes a weight associated to the component and is determined by considering its influence on the development process. In particular, $C=1$, means little influence; $C=2$, means medium influence; $C=3$, means strong influence.
3. For each script used to provide a functionality to manipulate document on the client side (i.e., in the web browser), count 1 entry .
4. For each application executed on the client side, count $C \cdot (1 \text{ entry} + 1 \text{ exit})$, where C denotes again a weight. The *entry* is considered to run it and the *exit* to show it.
5. For each server side interpreted script or compiled module used to produce a dynamic web page, count $1 \text{ entry} + 1 \text{ read} + 1 \text{ exit}$. In this case, a form allows users to input data and request a dynamic page (*entry*). The web server elaborates the input of the user through the server-side script or module (*read*) and produces a web page which is sent to the user (*exit*). Moreover, count an additional *read* if an access control is first performed and then the input is elaborated to generate the web page.
6. For each server side script modifying persistent data through the web server, count $1 \text{ entry} + 1 \text{ write} + 1 \text{ exit}$. The user inputs data through a form (*entry*), the data is written through the web server (*write*) and the result is shown to the user (*exit*). Count an additional *read* if an access control is first performed.
7. For each web page that contains confirmation, alert or error messages sent by the web server to the browser, count $1 \text{ read} + 1 \text{ exit}$.
8. For each reference to external applications such as commercial package, library routine, count $1 \text{ entry} + 1 \text{ exit}$.

Let us note that rules 5, 6, 7, 8 were specifically conceived to consider dynamic aspects of web applications, rule 2 refers to multimedia components and rules 1, 3, 4 take into account elements common to static web applications. In particular, the latter rules are analogous to the ones provided by Mendes *et al.* in [6] to measure hypermedia web applications. The sum of the identified data movements is expressed in terms of *CFSU*.

In the sequel we show the application of some rules. To this aim, let us consider the class diagram depicted in Fig. 1, which is referred to a web application designed for e-learning purposes and is concerned with the final test. The user requests the final test by specifying his/her data through the HTML form *UserRegistration* con-

tained in the client page *FinalTest*. The server page *UserIdentification* verifies whether or not the user is registered and the server page *TestCreation* prepares the form *TestForm* by using the information of the class *Test*. The user fills in the form by answering the questions and submits his/her test. Then, the server page *Scoring* interacts with the database and determines the score which is sent back to the user as an HTML page (i.e., *Score*). Moreover, the server page *DBUpdating* inserts the score into the database by using the user data contained in the object *Session*. The presence of the server pages *UserIdentification*, *TestCreation*, *Scoring* determines the application of rule 5, resulting in 9 *CFSUs*. Rule 6 is instead applied considering the server page *DBUpdating*, determining other 3 *CFSUs*. Finally, the presence of the static web page *FinalTest* which contains the HTML form *UserRegistration*, causes the application of rule 1, counting further 3 *CFSUs*. Thus, the total counting for the considered piece of design documentation is 15 *CFSUs*.

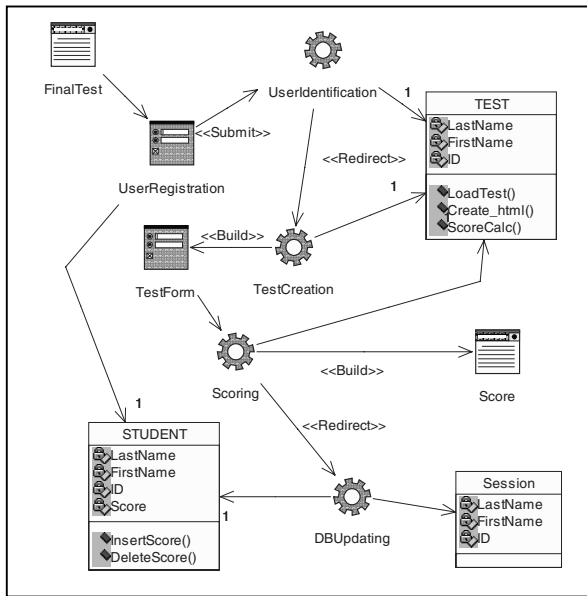


Fig. 1. The UML class diagram modelling the final test within an e-learning course

3 Empirical Evaluation

A statistical analysis has been performed to establish whether the proposed application of *COSMIC-FFP* can be used to predict the development effort of web based systems. We have exploited data coming from 32 web projects developed by students during the course on Web Engineering of two subsequent academic years. In both cases, the most skillful students were equally distributed among the groups in order to allow uniformity. Table 1 provides the descriptive statistics performed both for the variable *Effort (EFH)*, expressed in terms of person-hours, and the variable *COSMIC-FFP (C-FFP)*, expressed in terms of *CFSUs*.

Table 1. Descriptive statistics of *EFH* and *C-FFP*

	Obs	MIN	MAX	MEAN	STD. DEV.
EFH	32	62	172	117.6250	33.6699
C-FFP	32	84	833	346.4688	225.6862

In order to perform the empirical validation of the proposed method, we have applied an Ordinary Least-Squares regression analysis. The scatter plot of Fig. 2.a exposes a positive linear relationship between the variables *EFH* and *C-FFP*. From the Q-Q plot of Fig. 2.b we can observe that the residuals are normally distributed.

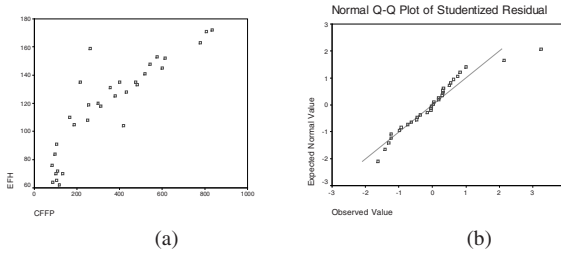


Fig. 2. (a)The scatter plot with *EFH* on the y-axis and *C-FFP* on the x-axis. (b) The Q-Q plot of the residuals.

From Fig. 3.a, we can observe that the linear regression analysis shows a high adjusted $R^2=0.763$, which indicates that 76.3% is the amount of the variance of the dependent variable *EFH* that is explained by the model. Moreover, we can observe a high *F* value and a low *p-value*, which indicate that the prediction is indeed possible with a high degree of confidence. The *p-values* and *t-values* for the corresponding coefficient and the intercept are reported in Fig. 3.b. The *p-values* give an insight into the accuracy of the coefficient and the intercept, whereas their *t-values* allow us to evaluate their importance for the generated model.

Prediction Model	adjusted R^2	R	Std Err	F	p-value
$EFH=0.131 * C-FFP + 72.254$	0.763	0.878	16.3978	100.699	0.000

(a)

	Value	Std. Err	t-value	p-value
<i>Coefficient</i>	0.131	0.013	10.035	0.000
<i>Intercept</i>	72.254	5.371	13.453	0.000

(b)

Fig. 3. The results of the OLS regression analysis for evaluating the *EFH* using *C-FFP*

In order to assess the acceptability of the derived effort prediction model, we have considered the *Mean of Magnitude of Relative Error (MMRE)* [4] over the 32 observations. The model exhibits an $MMRE = 0.1151$, which is less than 0.25, the threshold suggested by Conte *et al.* for an effort prediction model [4]. Moreover, we have evaluated the *prediction at level 0.25* [4], which has turned out to be 0.8438. According to Conte *et al.* a good effort prediction model should exhibit a value not less than 0.75, which is the case for the derived model.

4 Final Remarks

In this paper we have proposed an approach for estimating the functional size of dynamic web applications, exploiting the *COSMIC-FFP* method, during design phase. The results of the empirical analysis that we have carried out are encouraging, and suggest several research directions for future work. First of all, further analysis is planned for the assessment of the method. Indeed, the empirical evaluation provided in the paper has to be considered a preliminary analysis since the use of students' projects may threaten the external validity of the experiment and hence data coming from the industrial world are presently being collected, in order to obtain more reliable results. Such data will be also used to perform a comparative analysis with respect to other proposals, such as *Web Objects* [10,11], and to refine the current proposal by considering other features, such as the complexity of the data structure and algorithm behind the data movement. We finally plan to automate the application of the proposed rules on the class diagrams in order to integrate such functionality in a suitable CASE tool.

References

1. A.J. Albrecht, "Measuring Application Development Productivity", in *Proceedings of the IBM Application Development Symposium*, Monterey, 1979, pp. 83-92.
2. L. Baresi, S. Morasca, P. Paolini, "Estimating the Design Effort of Web Applications," in *Procs. IEEE Software Metrics Symposium*, Sydney, 2003, pp. 62-72.
3. J. Conallen, *Building Web Applications with UML*, Addison-Wesley, 1999.
4. D. Conte, H.E. Dunsmore, V.Y. Shen, *Software Engineering Metrics and Models*, The Benjamin/Cummings Publishing Company, Inc., 1986.
5. COSMIC: COSMIC-FFP Measurement Manual, vers. 2.2, <http://www.cosmicon.com>, 2003.
6. E. Mendes, N. Mosley, S. Counsell, "Comparison of Web Size Measures for Predicting Web Design and Authoring Effort", *IEE Proceedings-Software* 149 (3), 2002, pp. 86-92.
7. E. Mendes, I. Watson, C. Triggs, N. Mosley, S. Counsell, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications", *Empirical Software Engineering* 8(2), 2003, pp. 163-196.
8. M. Morisio, I. Stamelos, V. Spahos, D. Romano, "Measuring Functionality and Productivity in Web-based Applications: a Case Study", in *Procs. 6th Intern. Software Metrics Symposium*, Boca Raton, 1999, pp. 111-118.
9. T. Rollo, "Sizing E-Commerce", in *Procs. Australian Conference on Software Measurement*, Sydney, 2000.
10. D. Reifer, "Web-Development: Estimating Quick-Time-to-Market Software", *IEEE Software*, 17 (8), 2000, pp. 57-64.
11. M. Ruhe, R. Jeffery, I. Wiczorek, "Using Web Objects for Estimating Software Development Effort for Web Applications", in *Procs. IEEE Software Metrics Symposium*, Sydney, 2003, pp. 30-39.