

# Definition and Evaluation of a COSMIC Measurement Procedure for Sizing Web Applications in a Model-Driven Development Environment

*Silvia Abrahão<sup>a</sup>, Lucia De Marco<sup>b</sup>, Filomena Ferrucci<sup>b</sup>, Jaime Gomez<sup>c</sup>, Carmine Gravino<sup>b,\*</sup>, Federica Sarro<sup>d</sup>*

*<sup>a</sup>Department of Computer Science, Universitat Politècnica de València*

*<sup>b</sup>Department of Computer Science, University of Salerno*

*<sup>c</sup>Department of Languages and Information Systems, Universidad de Alicante*

*<sup>d</sup>CREST, Department of Computer Science, University College London*

---

## Abstract

**Context.** Model-driven development approaches facilitate the production of Web applications. Among them, the Object-Oriented Hypermedia method (OO-H) has been successfully used for the development of industrial Web applications. Similarly to other development approaches, it is important also in this context to put measures in place to support project managers in resource allocation, cost and schedule control, and productivity monitoring.

**Objective.** This motivated us to define a measurement procedure, named OO-HCFP, specifically conceived for OO-H Web applications based on COSMIC, a second-generation functional size measurement method.

**Method.** We present mapping and measurement rules devised to automatically derive size measures from OO-H models. We also carry out an empirical study to evaluate whether our proposed measurement procedure, OO-HCFP, is useful for estimating the effort needed to realise industrial Web applications developed with OO-H.

**Results.** The estimates obtained by using OO-HCFP are more accurate than those obtained by using other measurement approaches based on Function

---

\*. Corresponding author

*Email addresses:* `sabrahao@dsic.upv.es` (Silvia Abrahão), `ldemarco@unisa.it` (Lucia De Marco), `fferrucci@unisa.it` (Filomena Ferrucci), `jaime.gomez@ua.es` (Jaime Gomez), `gravino@unisa.it` (Carmine Gravino), `f.sarro@ucl.ac.uk` (Federica Sarro)

Points and design measures.

**Conclusions.** The proposed approach can be profitably exploited to size Web applications developed with OO-H. Based on our experience, we also provide some guidelines to support the formulation of COSMIC measurement procedures for other model-driven approaches.

*Keywords:* Web Applications, Model-Driven Development, Functional Size Measurement, COSMIC, OO-H method

---

## 1. Introduction

Several model-driven Web development approaches have been proposed in order to develop software at a higher level of abstraction by employing models and model transformations, e.g., W2000 [1], WebML [2], UWE [3], and OO-H [4]). These approaches support the construction of different views (i.e., models) of a Web application comprising at least a structural model, a navigation model, and a presentation model. Moreover, they provide tool support for the automatic generation of Web application source code from the specified models. Despite the fact that the majority of these approaches have been developed in academic contexts, some of them have also been used in industrial settings (e.g., OO-H, UWE, and WebML).

In general, the adoption of model-driven approaches in an industrial context poses new challenges which also result from the unknown impact of these practices on development effort and productivity [5, 6]. In fact, study [7] has highlighted that effort estimation employing COCOMO II, a widely used approach for traditional software systems, should be suitably adapted by re-interpreting various cost drivers in the context of model-driven engineering and taking into account the sizes of different artifacts. When model-driven approaches are used to develop Web applications, both the characteristics of the approaches and the specific characteristics of Web projects must be taken into account during several project management activities, such as resources allocation, costs and schedules planning and control, and productivity monitoring (see e.g., [8, 9, 10, 11, 12, 13]). These management activities crucially depend on the availability of size measures that are suitable for this kind of Web applications. In fact, a systematic literature review on Web resource estimation [14] confirms that research on effort estimation has focused on using size measures as predictors. The lack of accurate size estimations is recognized as the main cause of poor project management, and budget and

schedule overrun [15].

A fundamental problem in the context of model-driven Web application development is identifying a way to estimate size and effort starting from conceptual models. None of the ISO-standardized Functional Size Measurement (FSM) methods (e.g., Function Points Analysis, COSMIC) were designed by taking the specific features of Web applications and model-driven development into account. The existing FSM methods therefore need to be adapted or extended to deal with model-driven Web applications. These considerations motivate the research presented herein, which aims to provide an FSM procedure for Web applications developed using a model-driven development method (i.e., the OO-H method [16, 4]). We have selected this method for the following reasons : i) it has been applied during the development of several industrial Web applications ; ii) the availability of a comprehensive dataset of Web projects (including the corresponding conceptual models of the Web applications and the source code generated) that has been previously exploited for effort estimation purposes [17] ; iii) the flexibility of its CASE tool, which can be extended in order to automate size measurement procedures ; and iv) it can be considered as a representative method of the whole set of model-driven Web development methods [18].

We considered a functional size measurement procedure since FSM methods have obtained worldwide acceptance and allow software size measurement in terms of the functionality with which users are provided. The first FSM method was Function Point Analysis (FPA) [19], and several variants have since been defined (e.g., MarkII and NESMA) with the aim of improving size measurement or extending the applicability domain [20]. They are all part of the first generation of FSM methods, unlike COSMIC, which is considered to be a second generation FSM method owing to several distinguishing characteristics. In fact, COSMIC was the first FSM method developed to conform to the ISO/IEC14143/1 standard [21], it is based on fundamental principles of software engineering and measurement theory, it is applicable to business, real-time, and infrastructure software (or their hybrids) [22], and it is possible to define local extensions to the COSMIC measurement method for other kinds of software [23], including Web and Mobile applications [24, 25, 26].

With regard to Web applications, several studies have reported encouraging results in terms of both the applicability of the COSMIC method and its effectiveness for Web development effort estimation [9, 27, 28, 29, 30]. These considerations support the definition of COSMIC-based measurement

procedures for model-driven Web development, and in particular for OO-H. This research is further motivated by our previous results [31] which highlighted the poor performance of estimation models based on a first generation measure (OO-HFP) when compared to those based on some design measures computed by counting the modelling primitive describing the OO-H models.

In this paper we therefore present a COSMIC-based measurement procedure, denominated as OO-HCFP (i.e., Object-Oriented Hypermedia COSMIC Function Point) to size model-driven Web applications developed using the OO-H method. Mapping and measurement rules have been devised to automatically derive the size measure from the OO-H conceptual models. In addition, we present and discuss the results of an empirical study carried out to assess the effectiveness of the proposed measurement procedure for effort estimation. To this end, we have compared the prediction accuracy provided by OO-HCFP with those provided by OO-HFP and the OO-H design measures. The proposed measurement procedure can help project managers (and other team members without special training and certification in size estimation) to accurately estimate the size of a Web application developed following a model-driven development approach. The size measure is automatically obtained in an early phase of the Web development phase (conceptual model) and the results of our empirical study indicate that this measure can be considered as a suitable predictor of Web application development effort, since it provides more accurate estimations than a first generation FSM method and dimensional measures. We believe that these results are of interest to those Web companies that need to improve their size and effort estimation processes.

This paper enhances and extends a previous study [32] in which we provided only a brief description of OO-HCFP through the use of a running example. The main contributions of the present paper can be summarized as follows :

1. A more detailed and refined description of the COSMIC measurement procedure (i.e., OO-HCFP) ;
2. A plug-in carried out to automate the application of the measurement rules for OO-HCFP (and OO-H design measures), thus avoiding the ambiguity of interpreting the counting rules, subjectivity and mistakes in the counting process and the need for special training ;
3. An empirical assessment of OO-HCFP for effort estimation, by exploiting data from 30 Web applications developed using OO-H ;

4. A comparison of effort estimation accuracy as regards first generation functional size (i.e., obtained with OO-HFP) and second generation functional size (i.e., obtained with OO-HCFP);
5. A comparison of effort estimation accuracy as regards the proposed functional size (i.e., obtained with OO-HCFP) and various dimensional sizes (i.e., OO-H design measures);
6. A set of guidelines derived from the experience gained after formulating the proposed measurement procedure for supporting the definition of COSMIC measurement procedures that can be used with other model-driven approaches.

The paper is organized as follows. Section 2 outlines the research process we followed. Section 3.1 provides an overview of the OO-H method, and the main concepts of OO-HFP and OO-H design measures. Section 4 presents the design of the proposed measurement procedure (OO-HCFP) and its application to a running example. The design and the results of the empirical study performed to assess the effectiveness of OO-HCFP for effort estimation are described in Section 5. Section 6 focuses on the guidelines we propose in order to conceive COSMIC measurement procedures for other model-driven methods and the results of their application to the WebML approach. Related work is discussed in Section 7. Section 8 closes the paper discussing the implications of our work for practitioners, researchers and educator, and future work.

## 2. Research Process

This section describes the research process that we followed to design and evaluate the proposed COSMIC functional size measurement procedure. This process could be considered as an instantiation of the more general process suggested by Wieringa [33], which can be summarised as problem analysis, solution design, solution evaluation.

*Step 1. Problem Statement* : A fundamental problem in the context of model-driven Web application development is identifying a way to estimate size and effort starting from conceptual models. The existing ISO-standardized FSM methods were not designed by taking the specific features of Web applications and model-driven development into account. Therefore they need to be adapted or extended to deal with model-driven Web applications.

*Step 2. Research Goal* : The goal of our work is to derive an FSM procedure for Web applications developed using a model-driven development method. Since there exist various methods and it would be hard to derive a procedure that fits them all, during this step we used purposive sampling to identify a suitable model-driven development method that satisfies the following criteria :

- it is representative method of model-driven Web development methods ;
- it has been applied for the development of several industrial Web applications which are available to the researchers and have been previously exploited for effort estimation purposes ;
- its CASE tool can be extended in order to automate size measurement procedures.

Since the OO-H method meets all the above criteria we decided to focus our investigation on Web applications developed with this method.

*Step 3. Literature Review* : We reviewed the existing literature using snowballing in order to identify the main approaches used to size model-driven Web applications (note that a systematic literature review was out of the scope of this study). During this step we identified two main categories : one based on functional size measures and one based on design measures. A summary of the approaches found in the existing literature is reported in Section 3.2.

*Step 4. Design of the Measurement Procedure* : The measurement procedure we aim to design should be compliant to the process model for functional size measurement proposed by Jacquet and Abran [34] and the Guide to the Verification of FSM methods (ISO 14143-3) [35]. Therefore, we designed the OO-H measurement procedure by :

- defining the measurement procedure objectives (see Section 4.2 ;
- characterizing the concept to be measured (see Section 4.2.1 ;
- selecting the metamodels to represent the Web application to which the proposed functional size measurement procedure will be applied (see Section 4.2.1) ;
- defining mapping rules among metamodel elements, i.e. each OO-H concept to be mapped onto a COSMIC concept were defined by taking into account the COSMIC Software Context Model and the Generic Software Model. The output is shown in Table 1.
- defining the set of measurement rules in order to identify the four kinds of COSMIC data movement (1 Entry(E), Exit(X), Read(R),

Write(W)) in the OO-H conceptual models. The output is shown in Table 2.

In order to facilitate the reader understanding the application of the measurement rules we provide him/her with a running example which shows how to use the rules we designed. Moreover, the OO-HFP measurement procedure has been automated in VisualWADE by means of a plug-in which allow modelers to generate a size measurement report for a Web application once its conceptual model has been specified. We compared the estimation of an industrial web project performed by an expert on functional size measurement against the size measurement obtained by the plug-in. This allowed us to verify whether the value that is produced is the correct application and interpretation of the proposed mapping and measurement rules.

*Step 5. Evaluation of the Measurement Procedure* : This step concerns the use of the OO-HCFP measurement results in different types of models (e.g., productivity-analysis models, effort estimation models, schedule estimation models, budgeting models). In order to evaluate the measurement results we carry out an empirical study as detailed in the following :

- 5.1 Objective.** The objective of our study is to assess the usefulness of OO-HCFP measurement results for effort estimation.
- 5.2 Data Collection.** We use 30 Web applications developed by Web companies using the OO-H method. These were new developments for typical Web applications, e.g., cinema management, hospital management, content management, intranets etc., and were developed using from five to eight programming languages including JavaScript (DHTML/DOM), J2EE (JSP, Servlet, EJB), PHP, HTML, andSQL. The staff consisted of small teams of between one and five developers. We compared the data collected (e.g., average development method, number of staff on the development team) against the data collected in other similar effort estimation studies (see Section 5.1).
- 5.3 Threats to Validity** The validity of empirical studies can be biased by several factors, in this step we assessed the threats that could affect the validity of our study and plan some countermeasures, as detailed in Section 5.5.
- 5.4 Data Analysis.** We assessed the proposed functional measurement procedure by verifying whether the size obtained with the OO-HCFP measurement procedure is a good cost driver for estimating the development effort of OO-H Web applications by :

- Comparing the effort estimation accuracy as regards first generation functional size (i.e., obtained with OO-HFP) and second generation functional size (i.e., obtained with OO-HCFP);
- Comparing the effort estimation accuracy as regards the proposed functional size (i.e., obtained with OO-HCFP) and various dimensional sizes (i.e., OO-H design measures).

**5.5 Validation Method and Evaluation Criteria** This is related to the validation method and evaluation criteria used for validating the accuracy of the obtained effort estimation models. We applied a leave-one-out cross-validation procedure by splitting the dataset into training and validation sets, and used evaluation criteria based on the Absolute Residuals as well as statistical and effect size tests (see Section 5.3).

*Step 6. Interpretation and Report Writing* : The objective of this step is to interpret the results of the empirical study. In this step, measurements are collected which then are analysed and evaluated (see Section 5.4). We also defined a set of guidelines derived from the experience gained after formulating the proposed measurement procedure for supporting the definition of COSMIC measurement procedures that can be used with other model-driven approaches (see Section 6.1). Finally, we show how the guidelines can be applied in practice (see Section 6.2).

### 3. Background

In this section we describe the Object-Oriented Hypermedia model-driven development method and the existing measurement approaches for sizing model-driven Web applications.

#### 3.1. The Object-Oriented Hypermedia Method

OO-H method is a model-driven development method that provides the semantics and notations needed to develop Web applications [4, 16]. It also allows the automatic generation of source code through the use of model transformation techniques. The OO-H development process starts with the conceptual modeling phase using two platform independent models, namely a UML Class Diagram (UCD) and Navigational Access Diagrams (NADs), in order to specify the content and navigation requirements at a high level of abstraction. A set of transformation rules are subsequently applied to



these diagrams in order to automatically generate an Abstract Presentation Diagram (APD) and the source code of the corresponding Web application. In fact, the whole development process and the related transformation rules have been automated in the VisualWADE CASE tool [36]. It was developed using the Python language, which gives it a powerful capacity for introspection, or in other words the ability to query and manipulate objects and meta-objects during run-time. Moreover, the modular architecture of VisualWADE makes it easily extendable by means of plug-ins that permit the dynamic addition of new features.

The UCD and NADs are designed by exploiting the information derived from domain and navigation analysis. The UCD provides the domain information structure, i.e., the static part of a Web application. Each NAD represents information, services, and the navigation paths required to fulfill the requirements of the associated user navigation. Different NADs represent different navigational views, i.e., the different points of view of the various types of users who interact with the Web application. The whole set of NADs is conceived as the navigational model.

The set of modeling primitives for NAD is composed of 4 types of constructs : Navigational Classes (NCs), Navigational Targets (NTs), Navigational Links (NLs), and Collections (Cs). NCs are a subset of the domain classes, and are enriched with attributes and method visibility according to the users permission and navigation requirements. NTs are grouping elements that contain the NCs, NLs, and Cs, and collaborate in the coverage of each users navigation requirement. An NT may derive one or more abstract pages depending on the value of the property 'effect' of an NL. The value source (a light arrow) indicates that the information specified in the NT will be presented on the current abstract page, while the value target (a dark arrow) indicates that the information will be presented on a different abstract page. NLs define the navigation paths that the user can follow through the system. There are six types of NLs :

- I-Link (Internal Link), defining the path inside the same NT ;
- T-Link (Traversal Link), defining the path between NCs of different NTs ;
- R-Link (Requirement Link), representing points at the starting navigation point of each NT ;
- X-Link (eXit Link), representing points outside the boundary of the application ;
- A pair consisting of an S-Link and an Sr-Link (service and response

links), representing the services and the view after response. The parameters for an S-Link can be set using a specified mode : *hidden* and *constant* modes that do not require the user to introduce values ; *immediate*, which is realized by typing the required value ; *selection*, which requires a choice to be made from among a set of possibilities, and *navigation*, which implies navigation for the parameter selection.

OO-H additionally allows the use of the standard OCL to define constraints expressed by means of filters on each link.

Cs are hierarchical structures defined by NCs or NTs ; they group different NLs, thus providing access to different kinds of information. For more details about these modeling primitives, the reader is referred to [4].

An excerpt of the OO-H meta-model is represented in Figure 1 which highlights the relationships among the main primitives described above.

An example of OO-H conceptual models for a Web application named Task Manager is shown as follows and will be exploited in Section 4 to show the use of the proposed FSM procedure. These models are shown in Figures A.6, A.7, A.8, A.10, and A.9 at the end of the paper. The application manages the activities of the development process of a project, as specified by the UCD in Figure A.6, in which a user (i.e., the USERclass) is associated with certain tasks (i.e., the TASK class). Each task is composed of various pieces of information, such as start date, scheduled end date, priority, and so on (i.e., the attributes of TASK), and can own a set of external files (i.e., the FILE class). The project manager is the person responsible for organizing the activities into folders (i.e., FOLDER). A user can store comments (i.e., COMMENT) on the tasks and send messages (i.e., MESSAGE) to communicate with other users. It is also possible to save a daily report (DAILY\_REPORT) which contains information related to the tasks on which the user was working. All parties involved in the project are stored in the system as contacts (i.e., CONTACT). Figure A.7(a) shows the NAD for USER. The R-Link “Entry point User” is the entry point to the application. The first abstract page corresponds to the CUSTOMERS navigational class that instantiates USER : if the user already exists, a menu will be shown, which is represented by the restricted home collection and is linked to the four NTs identified. Each NT can later be exploded to show its related navigation path. As an example, Figure A.7(b) provides detailed information about the NT CONTACTS. USERs functions are represented by the following NLs : *allContacts*, which allows the user : to see the information about all contacts, *byInitial*, to search for a contact by providing an initial, and

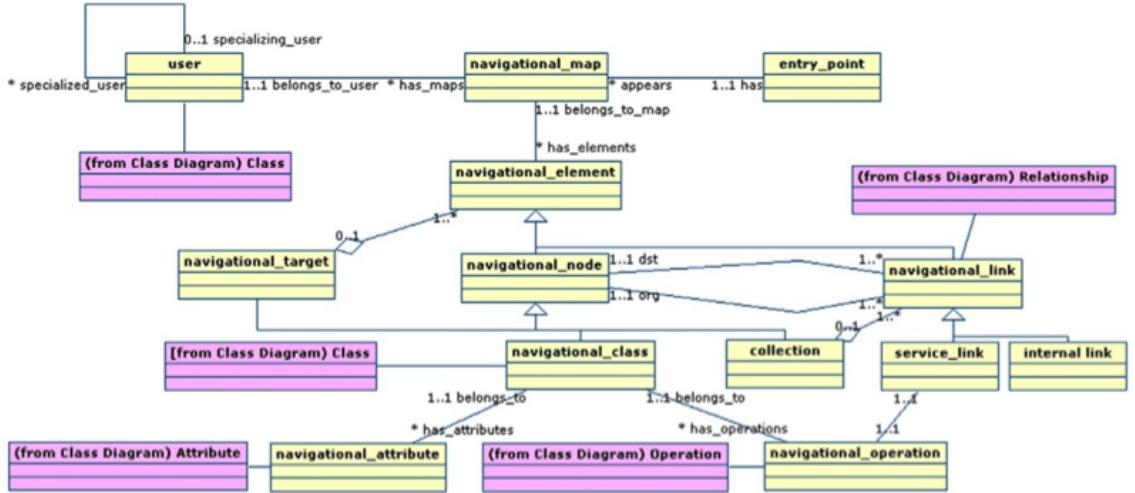


FIGURE 1: An excerpt of the OO-H meta-model [4].

*byString*, to search for a contact by providing a string.

Similarly, Figure A.8 shows the NT REPORTS. The “REPORT” collection represents the menu from which a user can access the daily reports in two ways (i.e., *byContent* or *byDates*) using the NC DAILY\_REPORT1. The NC DAILY\_REPORT2, meanwhile, allows access to the detailed report by selecting title (*LI47*) and author (*L41*). Other functions are the following : view all the reports in the DAILY\_REPORT4 navigational class (*LI46*), view all the users in the USER1 navigational class (*LI72*), or view the same days report in the DAILY\_REPORT3 navigational class (*Today*). Figures A.10 and A.9 show the other two Task Manager NADs (i.e., Projects and Notes).

### 3.2. Measurement Approaches for Sizing OO-H Web Applications

A study of the relevant literature allowed us to identify two main approaches with which to size model-driven Web applications : one based on functional size measures and the other based on design measures. Some FSM procedures have also been proposed (e.g., [37, 38]) and specifically conceived to map the modeling primitives of a model-driven approach onto the concepts of a functional size measurement method and to apply consistent measurement rules. Among them, Abrahão et al. [38] proposed OO-HFP for the sizing of Web applications developed using the OO-H method [4, 16]. The use of design measures is quite a different research approach as regards sizing

model-driven Web applications. These design measures take into account the modeling primitives that characterize the models of the specific model-driven approach. Examples of these measures are defined for the W2000 [1] approach in [39], in which the authors investigated whether the measures (e.g., Components, Clusters, and Slots) could be used to predict the design effort of Web applications developed using W2000. Following this approach, OO-H design measures have been proposed and compared with OO-HFP for the purpose of effort estimation [31]. In the following, we briefly describe the OO-HFP [17] and OO-H design measures [31] and provide a short report on the result of a previous study that compared them and further motivated the need for the measurement procedure described in this paper.

### *3.2.1. OO-HFP : A Function Points Measurement Procedure*

The OO-HFP measurement procedure was proposed in order to obtain the functional size of Web applications developed with the OO-H method, in terms of the IFPUG Function Points (FP, for short) [17]. FP represents the version of the FPA managed by the International Function Point Users Group (IFPUG).

FPA was the first FSM method to be proposed in literature; it was introduced by Albrecht of IBM in 1979 to measure a software product size in terms of the number of functions with which the end user was provided in the early development phases, meaning that the size measure was no longer related to the choice of technology adopted. The original formulation was extended several times, and in 2003 the non-adjusted part of the method (i.e., non-adjusted Function Points) was certified by ISO as an international standard (ISO/IEC 20926). FP identifies and classifies each function as External Input (EI), External Output (EO), External Inquiry (EQ), Internal Logical File (ILF), and External Interface File (EIF). Each function is then weighted depending on its type and level of complexity in agreement with standard values as specified in the Counting Practices Manual. The level of complexity is determined using the number of Data Element Types (DETs), File Types Referenced (FTRs), and Record Element Types (RETs) [40].

The OO-HFP measurement procedure is applied to the platform-independent models of an OO-H Web application (i.e., an UCD and a set of NADs) and exploits a specifically conceived set of mapping and measurement rules [17]. The mapping rules provide a mapping between the FPA concepts, as the counting scope and the boundary of the Web application, the data (ILF and EIF) and transactional (EI, EO, and EQ) functions, and the OO-H mode-

ling primitives. Moreover, a set of measurement rules is used to identify : 1) the DETs and RETs for each class in the UCD used to determine the complexity of a class or a class hierarchy ; 2) the DETs and the FTRs employed to determine the complexity of each method in the UCD ; 3) the DETs and RETs used to determine the complexity of an NT. Once DETs, RETs, and FTRs have been counted, the FPA counting rules are applied to classify the function complexity (low, average, high), to assign weights to the functions, and to aggregate the assigned values into an overall functional size value for the Web application.

### *3.2.2. OO-H Design Measures and Comparison with OO-HFP*

The OO-H conceptual models can be used as a basis on which to devise a set of design measures [31]. In particular, starting from the OO-H primitives we identified the measures shown in Table 1 that also specifies the model (i.e., UCD or NAD) where they are used. As for UCD, we identified the number of classes (CL), the number of associations (AS), the number of aggregations (AG), the number of compositions (CO), and the generalizations (GE). Concerning the NAD we identified seven measures as detailed in the following. Number of Navigational Targets (NT), the number of Navigational Classes (NC), the number of I-Link (IL), the number of T-Link (TL), the number of X-Link (XL), the number of R-Link (RL), the number of S-Link (SL), and the number of Collections (Cs). These measures were exploited to perform an empirical analysis aimed at comparing their effectiveness as regards estimating the OO-H Web application development effort with that of OO-HFP [31]. Upon employing 30 Web applications developed using OO-H (the same ones used in the empirical analysis described in this paper and described in Section 5.1), the analysis revealed that all OO-H design measures were positively correlated with development effort. However, the best estimation model, obtained by exploiting Manual StepWise Regression (MSWR) [41], employed only the IL and TL measures. Moreover, the results showed that the effort model based on IL and TL provided significantly better estimates than the OO-HFP model, thus confirming that the FPA may fail to capture some specific features of Web applications [27].

Although the aforementioned study suggests that design measures are better than functional size measures there exist some advantages in using functional size measurement methods. Indeed, functional size measures are considered clear for both customer organizations and supplier organizations : more functionality means more value, more effort needed and a higher costs

TABLE 1: OO-H design measures

Model	Measure	Description	Scale
UML Class Diagram (UCD)	CL	Number of classes	ratio
	AS	Number of Associations	ratio
	AG	Number of Aggregations	ratio
	CO	Number of Compositions	ratio
	GE	Number of Generalizations	ratio
Navigational Access Diagram (NAD)	NT	Navigational Targets	ratio
	NC	Navigational Classes	ratio
	IL	Number of I-Links	ratio
	TL	Number of T-Links	ratio
	XL	Number of X-Link	ratio
	RL	Number of R-Links	ratio
	SL	Number of S-Links	ratio
	Cs	Number of Collections	ratio

[20]. They have been widely used for planning support resources and budgets and for assessing the performance of software development project and benchmarking productivity. Moreover, another widely recognized improvement opportunity is related to size conversion for later phases in the life cycle. For example, traditionally, it is assumed that there is a linear and proportional relationship between the number of line of codes and the functional size [20].

This motivated research into other FSM procedures for OO-H that are able to capture aspects that correlate better with effort. It also contributed to the motivation of the research presented in this paper, i.e., analyzing, for the OO-H approach, the effectiveness of COSMIC, which has provided interesting results for Web applications in the context of development effort estimation (see e.g., [27, 9, 28, 29, 30]).

#### 4. Design and Application of the OO-HCFP Measurement Procedure

The goal of this work is to *analyze* a functional size measurement procedure *with the purpose* of automatically sizing Web applications modeled with OO-H *with regard to* their functional size *from the point of view of* both researchers evaluating how accurate the size estimations obtained at the conceptual model are, and project managers evaluating the possibility of

adopting the measurement procedure in their organization. This procedure, denominated as OO-HCFP [32], is compliant with the COSMIC method version 4.0.1 [22].

In the following subsections we first remind the reader of the main concepts of COSMIC and then present the OO-HCFP procedure.

#### 4.1. COSMIC

COSMIC is composed of a set of models, principles, rules, and processes that are applicable to the Functional User Requirements (FURs) of a given piece of software [22]. In particular, the Software Context Model introduces the principles and concepts needed to identify the FURs of the piece of software to be measured. The Generic Software Model has to be applied to the FURs to identify the components of the functionality that will be measured. Figure 2 illustrates the concepts specified by the Software Context Model and Generic Software Model at a high level of detail. The concepts of the Software Context Model are the following [22] :

- The **Purpose** of a measurement defines why a measurement is required.
- The **Scope** of a measurement selects the set of FURs to be included in a specific functional size measurement exercise.
- The **FURs** describe what the software will do in terms of tasks and services.
- A **Layer** is a partition resulting from the functional division of a piece of software architecture.
- A **Functional User** is a (type of) user that is a sender and/or an intended recipient of data in the FURs of a piece of software.
- The **Boundary** is defined as a conceptual interface between the software being measured and its functional users.

The following concepts of the Generic Software Model are applied to the FURs of each separate piece of software and are defined in [22] :

- A **Functional Process** is an elementary component of a set of FURs comprising a unique, cohesive and independently executable set of data movements. It is triggered by a data movement from a functional user that informs the piece of software that the functional user has identified a triggering event. It is complete when it has executed all that needs to be done in response to the triggering event. Each functional process consists of (a set of) sub-processes.

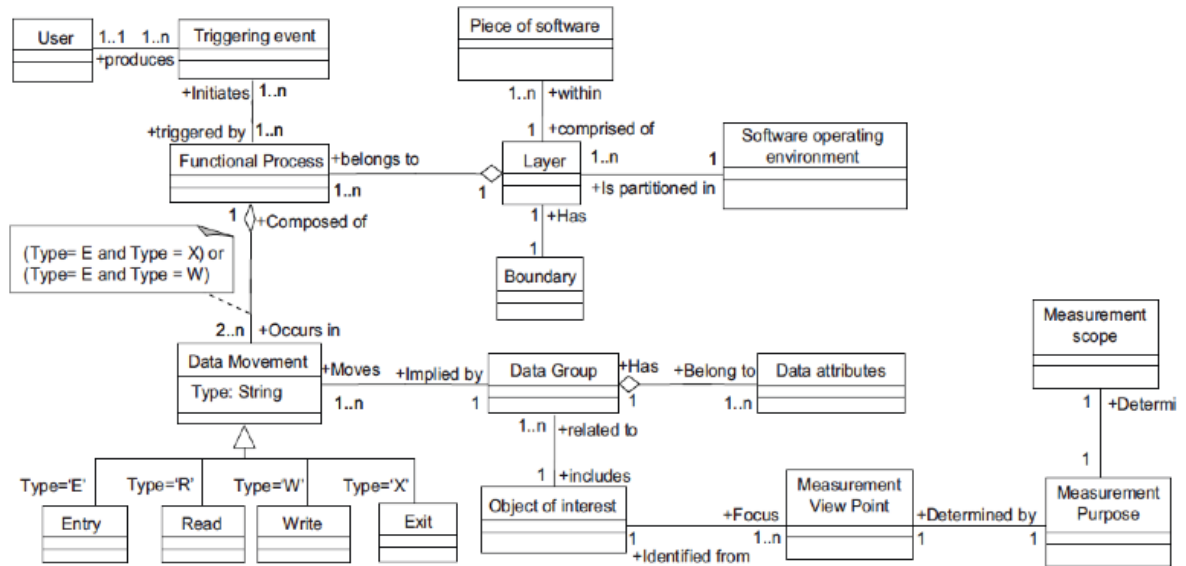


FIGURE 2: Meta-model for the COSMIC concepts [42]

- A **Functional Sub-Process** may be a data movement (i.e., Entry, Exit, Read, and Write).
- A **Triggering Event** is an event that causes a functional user of the piece of software to initiate (trigger) one or more functional processes.
- A **Data Group** is a distinct, non-empty, unordered, and non redundant set of data attributes describing a complementary aspect of the same object of interest.
- A **Data Attribute** is the smallest piece of information, within an identified data group, carrying a meaning from the perspective of the softwares FURs.

In the measurement phase, the data movements (i.e., Entry, Exit, Read, and Write) of each functional process have to be identified and used to obtain a size measurement for the software. These are defined in [22] as follows :

- An **Entry (E)** data movement moves a data group from a functional user across the boundary into the functional process where it is required.
- An **Exit (X)** data movement moves a data group from a functional process across the boundary to the functional user that requires it.
- A **Read (R)** data movement moves a data group from persistent



storage within each of the functional processes that require it.

- A **Write (W)** data movement moves a data group from within a functional process to persistent storage.

Each data movement is counted as 1 CFP, the COSMIC measurement standard. The size of a piece of software within a defined scope is therefore obtained by adding up the sizes of all the functional processes identified. For more details about the COSMIC method, readers are referred to the COSMIC Measurement Manual [22].

#### 4.2. OO-HCFP

In the following subsections, we present the set of mapping and measurement rules for the OO-HCFP measurement procedure and an example of their use in the OO-H Web application “Task manager”, presented in Section 3.1.

##### 4.2.1. Mapping rules

Table 2 provides a summary of the mapping rules conceived to identify the modeling primitives of the OO-H platform-independent models (i.e., UCD and the NADs) that contribute to the functional size of a Web application modeled with the OO-H method. They allow each OO-H concept to be mapped onto a COSMIC concept and were defined by taking into account the COSMIC Software Context Model and the Generic Software Model [22]. A description of these rules and the rationale behind them are shown as follows.

**Rule 1** is related to *FURs* that are specified in terms of the complete UCD and the NADs (i.e., the OO-H platform-independent models) of the Web application to be measured depending on the scope, such as those shown in Figures A.7, A.8, A.10, and A.9. The idea is that each NAD and the related classes of the UCD represent a FUR. **Rule 2** specifies that the whole Web application is considered as a layer<sup>1</sup>.

**Rule 3** specifies the scope of a Web application by limiting the functionality to be measured. We can decide to measure the whole Web application (i.e., full scope), by using the entire set of OO-H platform-independent models, or a part of it, by including the scope of a subset of NADs and the UCD classes related to the NADs considered (i.e., partial scope). In the case of a

---

1. A Web application obtained with OO-H follows a 3-tier architecture : client, server (business logic) and the database.

partial scope, we might be interested in the subset of NADs that represents the view of the navigation to specific users (one for each NAD).

The *functional users* are identified by considering all the classes in the UCD that represent a User type, also including the classes with « Actor » or « Legacy » stereotypes (**Rule 4**). In fact, COSMIC defines a ‘user’ as “anything that interacts with the software being measured” [22], and also includes other software components such as legacy systems. In the case of the diagrams in Figures A.6, A.7, A.8, A.10, and A.9, it is possible to apply Rule 4 to the User of the UCD class.

In order to identify the COSMIC *application boundary* of an OO-H Web application, we can trace an imaginary line in the UCD, putting the classes representing the functional users (e.g., User in Figure A.6) outside the boundary while the other classes (those included in the scope) are considered inside it. Of course, we have also included the NADs functions, which are related to the UCD classes considered (**Rule 5**), in the boundary definition.

**Rule 6** was conceived to map the COSMIC concept of a *functional process* onto a navigational target, since this element is the mechanism which groups the navigational elements that contribute toward realizing a functionality. Indeed, a navigational target may derive one or several abstract pages depending on the value of the property ‘effect’ of a navigational link (T-link). The value source (a light arrow) indicates that the information specified in the navigational target will be presented on the current abstract page, while the value target (a dark arrow) indicates that the information will be presented on a different abstract page.

**Rule 7** allows *functional sub-processes* to be identified from several OO-H elements, i.e., R-Links, I-Links, and S-Links, with related OCL constraints, which model the execution of an inquiry from the system to the persistent storage, and modulate the way in which the information is sent or obtained by a user. A sub-process can also be detected from the question mark on an I-Link, which indicates that the navigation is waiting for an input from the user.

A *triggering event* can be mapped onto a link (I-Link or T-Link) pointing to a starting navigation point (i.e., Cs or NC) inside an NT (**Rule 8**). The application of this rule to the NADs in Figures A.6, A.7, A.8, A.10, and A.9 allows us to identify the triggering events shown in Table 3.

*Data groups* were identified by first taking the UCD into account (**Rule 9.1**), since it represents persistent data managed by the Web application. Specifically, one data group is identified for each aggregation/composition

TABLE 2: OO-HCFP Mapping Rules

COSMIC concept	Rule	OO-H modeling primitives
FURs	1	A FUR is a navigational and data model of the application to be measured depending on the scope (i.e., NAD and related classes of the UCD). The set of all FURs are the complete UCD and all the NADs.
Layer	2	The whole Web Application
Scope	3	The whole set of OO-H platform-independent models (full scope) or a subset of NADs and a (related) subset of classes in the UCD (partial scope)
Functional user	4	Each class in the UCD representing a User type including the classes with Actor or Legacy stereotypes
(Inside the) Boundary	5	The classes of the UCD and related NADs within the established scope
Functional Process	6	Each abstract page derived from a navigational target of a NAD.
Functional Sub-Process	7	An R-Link, an I-Link, a question mark on an I-Link, an S-Link, or an OCL constraint on the (I or S)-link
Triggering event	8	A link (I-Link or T-Link) pointing to a Navigational Class/Collection (of a NAD) with an effect on Target
Data group	9.1	A class of UCD; an aggregation hierarchy; a composition hierarchy; an inheritance hierarchy (a data group for the superclass and a data group for the subclass if it includes new attributes)
	9.2	A Navigational Class that appears in the Navigational Target of the NADs and is considered in the FURs
	9.3	A Collection that appears in a Navigational Target of the NADs considered as FURs
Data attribute	10	An attribute of a class of the UCD

TABLE 3: Triggering Events for each NAD

NAD	Entry Point	R-Link	Collection	Navigational Class
NAD Level 0	Yes	LR1	HOME	SORTER : FOLDER
CONTACTS	Yes	LR5	CONTACT MENU	
PROJECTS	Yes	LR3		
REPORTS	Yes	LR4	REPORTS	
NOTES	Yes	LR6	NEW HOME	

hierarchy in the UCD since aggregation/composition represents ‘is part of’ relationships between classes. In addition, one data group is identified for each class that does not participate in an aggregation/composition relation. With regard to inheritance hierarchies, one data group is identified for the superclass and another is identified for the subclass if it includes new attributes. Moreover, other objects of interest can be identified for an OO-H Web application, namely the Web pages that are on an application server, waiting to be queried by a user. In the OO-H conceptual models, they can be identified by means of a set of NCs or Cs (**Rule 9.2** and **9.3**). Finally, **Rule 10** identifies data attributes from all of the UCD class attributes.

#### 4.2.2. Measurement Rules

According to the COSMIC measurement method, the functional size of the software is given by the sum of all the data movements for each functional process detected, where each data movement counts as 1 CFP. In order to assign a quantitative value that represents the size of an OO-H Web application, we therefore defined the set of measurement rules shown in Table 4. These rules allow us to identify the four kinds of data movement in the OO-H conceptual models. A description of these rules and the rationale behind them is shown as follows.

**Rules 11.1-11.6** were conceived to count the Entry (E) data movements. In particular, 1E is assigned to each R-Link (**Rule 11.1**). In fact, this kind of link carries the information needed by the system to move a Web page from a Web server to the client user interface and is triggered by a user who starts a functional process. By applying Rule 11.1 to the NADs in Figures A.6, A.7, A.8, A.10, and A.9, we identified 4E since there are 4 R-Links (*LR1*, *LR3*, *LR4*, *LR5*, and *LR6*) followed by the user to start the functional processes.

**Rule 11.2** states that 1E can also be identified for each I-Link. This kind of link carries information about the navigational class that the user wishes to navigate. The application of this rule to the diagrams in Figures 6, 7, 8, 9, and 10 produced 37E, since there are 37 I-Links connecting two NCs in the same NT.

**Rules 11.3, 11.4** and **11.5** allow 1E to be counted for each S-Link that introduces the parameters for the invocation of methods with the *hidden*, *constant*, or *immediate* mode. They are used to represent the interaction with a user that, for example, introduces information into a text area (immediate), or by clicking on an HTML link (hidden), or on an HTML button (constant). Rules 11.3 and 11.4 cannot be applied to the diagrams in Figures

TABLE 4: OO-HCFP Measurement Rules

Data Movement	Rule	OO-H modeling primitives
1 Entry(E)	11.1	1 R-link
	11.2	1 I-link
	11.3	1 S-link with the hidden mode
	11.4	1 S-link with the constant mode
	11.5	1 S-link with the immediate mode
	11.6	Question mark (?) for a single data group on an I-link
1 Exit(X)	12.1	1 I-link
	12.2	1 R-link
	12.3	1 Error message thrown by the system
	12.4	For each data group involved in an OCL constraint on an I-Link
1 Read(R)	13.1	1 R-link
	13.2	1 I-link
	13.3	For each data group involved in an OCL constraint on an I-link
	13.4	For each data group involved in an OCL constraint on an S-link. The data group must be different from the classes linked by the S-link
1 Write(W)	14	For each data group involved in the execution of an S-link

6, 7, 8, 9, and 10, while Rule 11.5 allows 9E to be obtained from the S-Links, thus introducing parameters for the invocation methods, with the interaction parameter set to *immediate mode*. The last rule for Entry considers the OCL filters on the I-Links labeled with a question mark (**Rule 11.6**), assuming that the interface is waiting for an external input, i.e., the user must introduce the information needed by the operation invocation. According to this rule, 5E can be obtained from Figures A.6, A.7, A.8, A.10, and A.9.

Four rules have been provided for Exit (X) data movements. **Rule 12.1** identifies 1X for each I-Link. These links retrieve the Web page needed to provide the user with the required functionality (the NC destination) from the Web server. Similarly, **Rule 12.2** counts 1X for each R-Link, since this kind of link retrieves the Web page needed by the user to initiate the required functionality from a Web server. Following the suggestion found in the COSMIC User Manual, we also introduced a rule to count 1X for each Error Message that occurs during the use of the system (**Rule 12.3**). **Rule 12.4** is related to the OCL constraints on an I-Link and counts 1X for each data group retrieved by this link and presented to the user via the NC (i.e., the target of the considered link). The application of these rules to the diagrams

in Figures A.6, A.7, A.8, A.10, and A.9 produces 43X from the 43 I-Links of the NADs that provide the user with the internal NCs (representing data groups) (by Rule 13.1). Another 5X can be identified from the 5 R-Links *LR1*, *LR3*, *LR4*, *LR5*, and *LR6* (by Rule 12.2), and a further 41X can be identified by Rule 12.4.

**Rules 13.1-13.4** are related to Read (R) data movements. **Rules 13.1** and **13.2** allow 1R to be counted for each R-Link and I-Link, respectively. In fact, these links move data from the Web server to the user to provide the Web page required, and in both cases the presentation of a correct Web page implies a read from the Web server in which these data groups are located. **Rules 13.3** and **13.4** take into account the OCL constraints associated with the links, since they imply further Read data movements. In particular, for each I-Link with an OCL constraint that requires further elaboration of information from persistent storage to be presented in the Web page, we count 1R for each distinct data group involved. Similarly, for each S-Link that obtains an OCL annotation which involves one or more classes that are different from the source and destination NCs, we count 1R for each distinct data group involved. The application of these rules to the diagrams shown in Figures 6, 7, 8, 9, and 10 produces 5R owing to the 5 R-Links *LR1*, *LR3*, *LR4*, *LR5*, and *LR6* (by Rule 13.1). The use of Rule 14.2 makes it possible to identify 43R, since the NADs contain 43 I-Links connecting two NCs present in the same NAD. Another 41R can be obtained from 41 I-Links with OCL constraints for the request for different data groups necessary for the functional processes (by Rule 13.3). We can also count 7R from the OCL constraints on the 7 S-Links that move data groups other than the NC source and destination (by Rule 13.4).

Finally, the Write (W) data movements are identified from the S-Links (**Rule 14**), and each S-Link indicates the execution of an operation. The application of this rule to the diagrams in Figures A.6, A.7, A.8, A.10, and A.9 allows us to count 11W from 11 S-Links whose attributes are involved in the execution of the related services.

By adding all the data movements obtained from the functional processes shown in Table 5 we therefore obtain a size of 258 CFP (62E, 89X, 96R, and 11W) for the Task Manager application.

TABLE 5: OO-HCFP for the Task Manager application

Functional process	Entry	Exit	Read	Write
NAD0	8	11	11	0
NOTES	3	3	3	0
REPORTS	7	29	30	2
PROJECTS	25	37	43	8
CONTACTS	9	9	9	1

## 5. Assessing the Effectiveness of OO-HCFP for Effort Estimation

We assessed the proposed functional measurement procedure by verifying whether OO-HCFP provides an accurate estimate of the effort required to develop OO-H Web applications. To this end, we carried out an empirical analysis based on an industrial dataset of 30 Web applications. As an estimation technique, we employed Ordinary Least Square Regression (OLSR), one of the most widely and successfully used estimation techniques. Moreover, we compared the accuracy of the estimates obtained when using OO-HCFP with those obtained when using OO-HFP and the OO-H design measures (see Sections 3.2.1 and 3.2.2).

The research questions we addressed were therefore :

- RQ1 : Is the OO-HCFP size measure able to estimate OO-H Web application development effort when used in combination with OLSR ?
- RQ2 : Are the effort estimates obtained using OO-HCFP superior to the effort estimates obtained using OO-HFP and the OO-H design measures ?

In the following subsections, we present the design of the empirical study carried out to address the research questions shown above. We start by illustrating the dataset employed and the results of the OO-HCFP measurement procedure applied to these systems (Section 5.1). We then describe the OLSR estimation technique used to obtain the effort predictions (Section 5.2), after which we describe the cross-validation method and the evaluation criteria employed to assess the estimates (Section 5.3). The results of the empirical analysis are reported and discussed in Section 5.4. Finally, the section concludes with a discussion regarding the validity of the empirical study (Section 5.5).

TABLE 6: Descriptive statistics for the 30 Web applications

Variable name	Min	Max	Mean	Median	Std. Dev.
OO-HCFP	20	4403	543	142	974
OO-HFP	30	2193	391	203	512
CL	1	71	18	10	21
AS	0	65	14	8	19
NC	2	408	59	20	100
NT	1	64	11	5	15
IL	4	671	88	26	148
TL	0	432	41	9	87
RL	1	38	8	5	10
SL	0	411	31	8	76
Cs	1	50	13	7	14
Effort	16	3644	623	123	1123
DevTeam	1	5	3.23	3	1.521
NumLanguages	5	8	5.19	5	0.654
ExpProgLang	4	8	5.61	5	1.453

### 5.1. Dataset

We employed 30 Web applications developed by a Spanish Web company using the OO-H method. These were new developments for typical Web applications, e.g., cinema management, hospital management, content management, intranets etc., and were developed using from five to eight programming languages including JavaScript (DHTML/DOM), J2EE (JSP, Servlet, EJB), PHP, HTML, and SQL. The staff consisted of small teams of between one and five developers. Table 6 shows the summary statistics for the variables employed in our empirical study, namely OO-HCFP, OO-HFP, and each of the OO-H design measures. Note that the full data collected cannot be provided for reasons of confidentiality. The sizes of these measures were obtained by using the VisualWADE tool [36], which has been extended with a plug-in in order to automate the application of the measurement rules for OO-HCFP, OO-HFP, and OO-H design measures. This avoids the ambiguity of interpreting the counting rules and the need for special training to count these measures in an accurate and repeatable manner. Moreover, Figure 3 shows the boxplots of the variables involved in our study so as to show the distributions of their values. Note that the box length and tails of boxplots of OO-HCFP and Effort, along with their medians, are quite close.

Table 6 also shows the descriptive statistics of the variables Effort (actual



effort in person/hours for building and testing a platform-independent model (UML Class Diagram and Navigational Access Diagrams), DevTeam (number of staff on the development team), NumLanguages (number of programming languages), and ExpProgLang (the development team’s experience of the programming languages (months)). Specifically, the actual effort includes the effort of gathering the requirements, modeling (i.e., the specification of the UML class diagram and NADs), verification of the models (e.g., refinement of the Abstraction Presentation Model (APD) for changing widgets position, adding texts, modifying widgets, changing the user interface design), code generation and testing. These activities are performed iteratively until the development is completed and the web application is deployed. We would like to highlight that the testing effort can take an important portion of the development effort, as acknowledged by other studies. For instance, Meyerhoff et al. [43], acknowledged that the preparation of tests for a Web-based system involves considerably more effort than conventional testing. Mills [44] acknowledged that testing web applications involve the application of both static testing techniques applied to a web page (e.g., functionality of hyperlinks and images, consistency of presentation elements) and dynamic techniques (e.g., server-side page tests, client-side/browser-page tests, transaction testing, non-functional testing). In the case of OO-H, the testing effort also included the effort related to the adaptation of the generated code or the manual coding of certain functions that were not automatically generated by the Visual Wade Tool.

The median for OO-HCFP, OO-HFP and Effort suggests that most of the projects were small as regards their size and duration. Note that these applications were developed following a model-driven development approach in which the source code was generated automatically using VisualWADE, hence their short duration. The average development effort made for a Web application (623 person/hours (ph)) is similar to that found in other related studies (883 ph) [8]. In terms of dispersion, the data collected has a higher dispersion (1123 ph) than in the study of Ruhe et al. (710 ph).

We are aware that the modeling phase implies an effort and require specialized expertise. However, when comparing OO-H with other similar model-driven development approaches (e.g., WebML, UWE), we believe that OO-H does not impose an effort beyond the expected modeling effort for this type of modeling approaches. It fits within the type of models (i.e., UML class/structural model, navigation model) that are usually built for this type of applications. The benefit here is that the modeler can focus on his/her

daily modeling activities and the sizing and effort estimation is automatically performed (OO-HCFP was automated in VisualWADE). This avoids the ambiguity of interpreting the COSMIC mapping and counting rules and the need for special training for using COSMIC in an accurate and repeatable way. Nevertheless, as future work, we plan to carry out an empirical study similar to the one performed by Jolak et al. [45] for assessing the effort involved when modeling with OO-H.

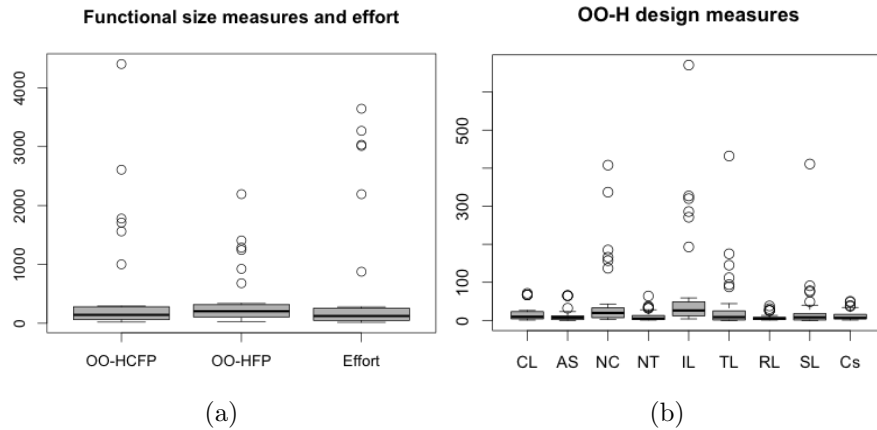


FIGURE 3: Boxplots of the variables employed in our empirical study

The number of staff on the development team (between 1 and 5) is similar to that of other studies published in the field of Web engineering [46, 11]. These studies show that the typical size of a Web development team is small. Regarding team composition, 11 of the projects were developed by a team composed of one project manager, one modeler, two programmers, and one tester. In some cases (5 of the projects), there was an overlap of staff (the same person played the roles of modeler and tester or modeler, programmer and tester). The remaining 19 projects have been developed by a team composed of one member who played the roles of modeler, programmer, and tester.

The data additionally indicated that the Web applications were mainly generated automatically using the following programming languages : Javascript (DHTML/DOM), PHP, HTML, SQL, and Python. Other projects also included the use of J2EE (Servlet, EJB). In terms of experience, the development team had an average experience of UML/OO-H of 3.84 years. The development teams modeling experience was ranked by the project manager on a scale of 1 - 5 (i.e., 1 = beginner, 2 = low, 3 = medium, 4 = advanced,

5 = expert). Finally, the development teams experience of the programming languages used was between 4 and 8 months.

### 5.2. Estimation technique

Ordinary Least Square Regression (OLSR) is a statistical technique that explores the relationship between a dependent variable and one or more independent variables, providing a prediction model described by an equation of the following type

$$y = b_1x_1 + b_2x_2 + \dots + b_nx_n + c$$

where  $y$  is the dependent variable,  $x_1, x_2, \dots, x_n$  are the independent variables,  $b_i$  is the coefficient that represents the amount variable  $y$ , which changes when variables  $x_i$  change by one unit, and  $c$  is the intercept. In our empirical study, for OO-HCFP and OO-HFP we used OLSR to obtain a linear regression model that uses the variable representing the effort as a dependent variable and the variable denoting the size measure employed as an independent variable. Once the prediction model has been constructed, the effort estimation for a new Web application is obtained by sizing the application in terms of the size measure chosen (e.g., OO-HCFP) and using this value in the model obtained. With regard to OO-H design measures, we built a linear regression model by applying Manual StepWise Regression (MSWR) and employing all the OO-H design measures considered. In particular, we exploited the technique proposed by Kitchenham [41], which allowed us to compute a linear regression analysis in steps. The estimation model is obtained by adding the independent variable with the highest correlation to the dependent variable in each step, taking into account all the variables currently in the model. The idea underlying this procedure is to select the best fitting model. We did not take into account other estimation methods, such as machine learners, e.g., [47, 48], search-based approaches [49, 50, 51, 52, 53, 54, 55, 56, 57] or their combination, e.g., [58, 59, 60], since our focus was to compare FSM methods rather than specific techniques.

Several indicators can be considered when evaluating the goodness of fit of a regression model. Of these,  $R^2$  shows the amount of variance of the dependent variable that can be explained by the model related to the independent variable. Other useful indicators are the  $F$  value and the corresponding  $p$  - value (denoted by  $SignF$ ). A high  $F$  value and a low  $p$ -value ( $< 0.05$ ) denote a high degree of confidence in the prediction. Moreover, the

stability of each effort estimation model was verified by analyzing the possible influential outliers following the procedure proposed in [61], which is based on the use of Cook’s distance and residuals (by exploiting plots). In particular, residual plots are able to show residual vs. fitted values and the goal is to determine whether the residuals are random or normally distributed. The Cook’s distance are then used to verify the presence of influential observations. With this procedure, any observation having distance greater than  $3x4/n$  (where  $n$  represents the total number of observations in the training set) is removed from the data analysis. While those observations in the training set with a Cook’s distance less than  $3x4/n$  but higher than  $4/n$  are removed to test the model stability, which is done by observing the effect of their removal on the model. If the model coefficients remained stable (i.e., they present similar values to the ones in the previous model) and the adjusted  $R^2$  improved, the highly influential projects were retained in the data analysis.

### 5.3. Validation Method and Evaluation Criteria

We applied a cross-validation by splitting the dataset into training and validation sets. Training sets are used to build estimation models and validation (or test) sets are used to validate the models obtained. In particular, we used a leave-one-out cross validation procedure, which means that the original dataset was divided into  $n$  (i.e.,  $n$  is the size of the original dataset) different subsets of training and validation sets, in which each validation set contained one observation [62].

As evaluation criteria we exploited the Absolute Residuals (AR), defined as  $|Effort_{actual} - Effort_{predicted}|$ , where  $Effort_{actual}$  and  $Effort_{predicted}$  are the observed and the predicted effort, respectively. In order to obtain summary measures so as to compare different estimation approaches, we employed both Median and Mean of AR (MdAR, MAR) as suggested in recent studies [63, 64, 57, 65]. We also report other summary measures, namely the mean and median of MRE, Pred(25), and the mean and median of EMRE, which have been widely used for effort estimation [66]. This has been done only for completeness (i.e., to allow for a comparison with previous research done in this context) but we do not use these measures for the assessment and comparison of the effort estimates obtained since their use has been strongly discouraged in previous work (e.g., [67]).

Furthermore, we performed statistical tests to establish whether one estimation model provided significantly better estimates than another by consi-

dering absolute residuals [68]. In particular, we performed the Wilcoxon signed rank test (or the unpaired version, i.e., Mann-Whitney U test) to verify the following null hypothesis : “the two populations of absolute residuals considered have identical distributions”. This kind of test is used to verify the hypothesis that the mean of the differences in the pairs is zero. We applied the Wilcoxon test since the absolute residuals obtained were not normally distributed as suggested by the Shapiro Wilk test [69]. Note that the results of the tests applied were considered as statistically significant at  $\alpha=0.05$  (i.e., at a 95% confidence level).

In order to have also an indication of the practical/managerial significance of the results, we verified the effect size. Effect size is a simple way of quantifying the standardized difference between two groups [70]. 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 [70]. In the empirical software engineering field, the magnitude of the effect sizes measured using the Cliffs  $d$  can be classified as follows : negligible ( $d < 0.147$ ), small (0.147 to 0.33), medium (0.33 to 0.474), and large ( $d > 0.474$ ) [70].

#### 5.4. Results

In this section, we present the results<sup>2</sup> of the empirical study by first reporting the evaluation performed to assess the accuracy of the OO-HCFP measure as regards estimating the development effort (RQ1). We then provide the results of the comparison with the OO-HFP and OO-H design measures (RQ2).

In order to apply OLSR, we first verified the underlying assumptions, i.e., the existence of a linear relationship between the independent and the dependent variables (*linearity*), the constant variance of the error terms for all the values of the independent variable (*homoscedasticity*), and the normal distribution of the error terms (*normality*). Since the homoscedasticity and normality assumptions of the residuals were not verified, we used the natural log transformation in an attempt to approximate the values to a normal distribution [61]. We obtained new variables, e.g., LnEffort, LnOO-HCFP, LnOO-HFP, LnIL, and LnTL, which represent the transformed variables Effort, OO-HCFP, OO-HFP, IL, and TL, respectively. In addition, whenever a

---

2. The results presented in this section were obtained using the statistical software R.

TABLE 7: The results of the OLSR analysis

Var.	R <sup>2</sup>	Adj. R <sup>2</sup>	Std.Err.	F	Sign. F
OO-HCFP	0.97	0.97	0.22	790.1	<0.01
OO-HFP	0.89	0.88	0.4	220	<0.01
IL, TL	0.97	0.97	0.26	498.4	<0.01

TABLE 8: The results of the leave-one-out cross validation

Var.	MdAR	MAR.	MMRE	MdMRE	Pred(25)	MEMRE	MdEMRE
OO-HCFP	16.83	59.88	0.19	0.16	0.62	0.19	0.17
OO-HFP	46.13	229.79	0.46	0.38	0.40	0.50	0.28
IL and TL	18.19	136.04	0.24	0.18	0.60	0.24	0.20

variable needed to be transformed but had zero values, the natural logarithmic transformation was applied to the variables value after adding one. With regard to the models based on OO-H design measures, the application of the MSWR procedure described in Section 5.2 revealed that the best fitting model obtained was that which employed only IL and TL as independent variables.

Table 7 show some statistics related to the estimation models based on OO-HCFP, OO-HFP, and OO-H design measures. The three models are characterized by a high R<sup>2</sup> value (0.97, 0.89, and 0.97, respectively), a high F value (790.1, 220, and 498.4, respectively), and a low Sign F (< 0.01), indicating that the prediction is possible with a high degree of confidence.

The final estimation models, when transformed back to the raw data scale, give the following equations :

$$\begin{aligned} \text{Effort} &= \text{OO-HCFP}^{1.11} 0.47 \\ \text{Effort} &= \text{OO-HFP}^{1.28} 0.18 \\ \text{Effort} &= \text{IL}^{0.79} \text{TL}^{0.28} 5.87 \end{aligned}$$

The analysis of the indicators used to evaluate the goodness of fit of the estimation model based on OO-HCFP suggests that we can positively answer the first research question, i.e., *the OO-HCFP size measure is able to estimate Web application development effort when used in combination with OLSR.*

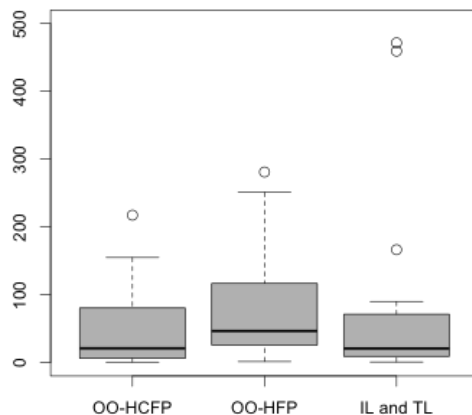


FIGURE 4: Boxplot of absolute residuals

With regard to the comparison between estimates achieved using OO-HCFP and those obtained by employing OO-HFP and OO-H design measures, the boxplots of absolute residuals in Figure 4 show that the box length and tails of OO-HCFP are less skewed than those of OO-HFP, while they are quite close to those of IL and TL. Moreover, an analysis of the corresponding medians, whose values are also reported in Table 8 (i.e., MdAR), suggests that the MdAR value of OO-CFP is closer to zero than the MdAR value obtained with OO-HFP and IL and TL measures. Thus, OO-HCFP provided better estimates than OO-HFP and design measures (i.e., IL and TL).

To better analyze the differences between the achieved estimates, we computed the mean percentage differences<sup>3</sup>.

The analysis of the descriptive statistics reported in Table 8 reveals that the mean percentage difference of the MdAR values obtained with OO-HCFP and OO-H design measures IL and TL is 23%. In the case of OO-HCFP and OO-HFP the difference is 175%. As for the comparison in terms of MAR values, the mean percentage difference between OO-HCFP and OO-H design measures is 134%, while in the case of OO-HCFP and OO-HFP the difference is 295%.

We also applied the Wilcoxon test to verify whether the difference highlighted is statistically significant. The results suggest that the estimates ob-

---

3. Given two values  $a$  and  $b$ , the mean percentage difference of  $a$  and  $b$  is computed as  $(a - b)/b * 100$

tained with OO-HCFP were significantly better than those obtained with OO-HFP (the p-value is 0.02) with a medium effect size ( $d=0.33$ ), while there was no statistically significant difference between the estimates obtained with OO-HCFP and IL and TL (the p-value is 0.12) with a small effect size ( $d=0.19$ ).

The answer to the second research question is therefore that *the effort estimates obtained using OO-HCFP are better than those obtained using OO-HFP and design measure*. The difference is also statistically significant when comparing OO-HCFP and OO-HFP.

### 5.5. Threats to Validity

The validity of empirical studies can be biased by several factors. In the following we discuss the threats that could affect the construct, internal, conclusion, and external validity of our study according to the best practises for Empirical Research in Software Engineering [71, 72, 73, 74].

With regard to the construct validity, the collection of information related both to the size and the actual effort is a crucial aspect. Size values in terms of OO-HCFP, OO-HFP and OO-H design measures were automatically obtained by means of VisualWADE plug-ins. This obviously reduces the possibility of manual error and rules out misunderstandings. Nevertheless, it introduces a possible threat to validity caused by an erroneous implementation of the mapping and measurement rules. This threat was mitigated by carefully testing the plug-in using the results of the manual application of the rules as an oracle. Furthermore, one of the authors supervised the procedure employed by the company involved to collect the information used for the empirical analysis in a controlled and uniform fashion. In order to maximize accuracy, both an interview format and the self-report format were used. The procedure consisted of using an excel file to collect a range of variables for each of the 30 applications. The project manager at the company filled in the excel file and one of the authors performed interviews to ensure that the data collected was accurate.

Various factors should be taken into consideration for the internal validity : the participants' authoring and designing experience, and the reliability of the data and lack of standardization [71]. The developers involved in the study were professionals who worked at the software company. No initial selection of the developers was carried out, so no apparent bias was introduced. The Web applications employed in the empirical study were designed and developed using a model-driven Web development process based on the OO-H



method and VisualWADE [36], of which the developers had previous experience, and it is consequently possible to exclude confounding effects resulting from the methods employed and the tools used. With regard to the reliability of the data and lack of standardization, the OO-HFP and OO-HCFP size measures and the OO-H design measures were obtained automatically using VisualWADE plug-ins implementing the two measurement procedures. The actual effort was collected using the same questionnaire for all the Web applications and the developers were instructed on how to use it in order to provide the information required correctly.

In the case of the conclusion validity, we carefully applied the estimation methods and the statistical tests, and verified all the assumptions required.

With regard to external validity, the Web applications employed in our empirical analysis were developed by a single software company. However, the results achieved can be generalized to Web development companies that follow a model-driven development process based on the OO-H method and are characterized by a similar industrial context. In general, the results obtained in an industrial context might not hold in other contexts that can be characterized by different features [62]. In addition, OO-H is not a standard for model-driven Web development. Unfortunately, at the time that this work was conceived there was no standard notation in the field. In 2013, the Interaction Flow Modeling Language (IFML) was adopted as a standard by the Object Management Group. As discussed by Brambilla and Butti [75], IFML can be viewed as the consolidation of the Web Modelling Language (WebML), which was defined and patented about 15 years ago as a conceptual model for data-intensive Web applications. Nevertheless, note that even though there is now a recently proposed standard for the model-driven development of Web applications, there is no data available as regards Web projects developed using IFML that allow a size and effort estimation study like that provided in this paper to be performed.

## **6. Guidelines for Defining a COSMIC Measurement Procedure for Model-Driven Approaches**

The aim of this section is to understand to what extent OO-HCFP can be extended/adapted to other model-driven approaches. It is worth noting that a single COSMIC-based measurement procedure that is able to work for any model-driven approach cannot be devised owing to the fact that each of them is characterized by specific concepts and modeling aspects for which specific

mapping and measurement rules must be formulated. Nevertheless, we exploited the lesson learned in order to derive guidelines with which to support the definition of COSMIC measurement procedures for other model-driven Web application development approaches. In the following subsection, we present how these guidelines have been conceived and the results of their application as regards identifying the mapping and measurement rules for a COSMIC measurement procedure for the Web Modeling Language (WebML), which is one of the most successful model-driven approaches for developing Web applications [76].

### *6.1. Derived Guidelines*

In order to define a COSMIC measurement procedure it is necessary to define a set of mapping and measurement rules, as has occurred in previous works such as [32, 37, 77, 78, 79]. Our intention was, therefore, to define a set of guidelines that should allow the mapping and measurement rules for a specific model-driven approach to be derived.

As highlighted in Section 4.2.1 for OO-HCFP, it is first necessary to identify the FURs. Since COSMIC [22] affirms that the FURs of a “new developed” application reside in the software artifacts, those artifacts must be intended as the analysis and design models employed in the model-driven context. In particular, among the models conceived in each approach, we have to identify those to be considered for the measurement. We therefore suggest focusing on the structure and navigation models, because they abstract the managed data and the users navigations, respectively (Rule 1 in Table 9). The models related to the layout and graphical appearance, i.e., those related to presentation, cannot be considered because they are useless for identifying the COSMIC concepts. Indeed, NADs and UCD were employed for the OO-H method, while the APDs models were excluded.

Once the proper design models have been selected, they can be used to identify the primitives that represent the COSMIC concepts, e.g., functional processes, data groups, and data movements. Please note that the identification of the software layers can be avoided, as most of the Web Applications built following a model-driven approach are Management Information System (MIS) and are composed of only one COSMIC layer (see Rule 2 from Table 9) : the Application layer. Since the identification of functional processes, data groups, and data movements may vary depending on the scope, we should decide to which measure the application is to be extended. Indeed, the concept of “scope” in COSMIC determines the portion of the software

to be measured, delineating the pieces of software that will be included and excluded. In the case of OO-H, we set the scope as the whole application by considering the OO-H PIM or a part of it by choosing a subset of the NADs and a (related) subset of classes in the UCD. In order to extend this rule to other model-driven approaches, we can therefore exploit Rule 3 in Table 9, which suggests taking into account all the design models to measure the whole application, or a subset of them if we wish only to measure part of the application.

Once the scope has been set, we can identify all the functional users and the functional processes from the design models. In the case of OO-H, the classes in the UML Class Diagram that represent a User type (also including the classes with `<< Actor >>` or `<< Legacy >>` stereotypes) are used to identify functional users. In order to generalize to any model-driven approach we therefore conceived Rule 4 in Table 9, which suggests focusing on the elements of the structure model that represents a User type. With regard to functional processes, since they are initiated after a triggering event has been launched by a functional user, the navigational models have to be decomposed by identifying which portions of them involve a single functionality. Moreover, each portion must include all the operations needed to accomplish the functionality requested by the triggering event. Note that a feasible decomposition must ensure that all the functional processes identified are always triggered by an event, i.e., no functional process can be identified if we do not find a design primitive that represents a sort of triggering/starting point in the navigational model (see Rules 6 and 8 in Table 9). Please recall that in the case of OO-H, a Navigational Target of a NAD represents a functional process if we can find an R-Link that represents the triggering event.

Rule 9 in Table 9 suggests that the identification of the data groups involves the modeling primitives from the structure model because they represent the static part of the system and are considered to be the data managed by the application. Furthermore, they are connected to each other via relations and are composed of atomic information (i.e., attributes). In addition, the navigational modeling primitives that represent the Web pages in the Web server can represent data groups since they are also moved by the data movements. In the case of OO-H we therefore identified data groups from each Class of the UCD for the persistent data, each Navigational Class and each Collection of the Navigation Target for the navigation model.

With regard to the identification of functional sub-processes, in OO-H this step requires considering the NT design components, i.e., the R-Links,

TABLE 9: General guidelines

COSMIC concept	Rule	Modeling primitives
FURs	1	The set of Functional User Requirement comprises all the navigation and structure models that abstract the users navigation and the data managed by the application, respectively
Layer	2	The whole Web application modeled is considered to be a single COSMIC layer, i.e., the Application layer
Scope	3	Overall scope : the complete set of navigation and structure models included in the FURs; Partial scope : a subset of the navigation models together with the related (portion of the) structure model
Functional User	4	The elements of the structure model that represent a User type
(Inside the) Boundary	5	The elements of the navigation and structure models selected according to the established scope
Functional Process	6	A portion of the navigation model included in the measurement scope that represents a complete set of operations that performs a functionality and that is commenced by an event triggered by a Functional User
Functional Sub-Process	7	A single element of a navigational model that represents an operation included into a Functional Process
Triggering Event	8	An element of the navigational model representing a Functional Process that begins its execution
Data Group	9	An element of both the navigation and the structure model that abstracts an entity processed by the application; it includes all the attributes that describe a single entity
Data Attribute	10	A piece of information related to a single Data Group that describes a characteristic of it
Entry Data Movement	11	An element from the navigation models previously identified as a functional sub-process; in particular, a navigational element which abstracts the passage of a single data group from an element of the structure model that represents a Functional User to an element of a navigational model that represents a Functional Process
Exit Data Movement	12	An element from the navigation models previously identified as a functional sub-process; in particular a navigational element which abstracts the passage of a single data group from an element of a navigational model that represents a Functional Process to an element of the structure model that represents a Functional User
Read Data Movement	13	An element from the navigation models previously identified as a functional sub-process; in particular a navigational element which abstracts the passage of a single data group from an element of the structure model that represents a Data Group to an element of a navigational model that represents a Functional Process
Write Data Movement	14	An element from the navigation models previously identified as a functional sub-process; in particular a navigational element which abstracts the passage of a single data group from an element of a navigational model that represents a Functional Process to an element of the structure model that represents a Data Group

I-Links and their question mark inputs, S-Links, and OCL constraints on the I and S Links. This signifies that, for any model-driven approach, the functional sub-processes can be identified by analyzing the navigation models design primitives exploited to obtain the functional processes (see Rule 7 in Table 9).

Once functional processes and sub-processes have been identified, a set of measurement rules should be defined in order to identify data movements according to the definitions of Entry, Exit, Read and Write provided by COSMIC. We suggest that these rules can be drawn by exploiting the Rules from 11 to 14 reported in Table 9. It is worth mentioning that each design element identified as a sub-process can potentially represent more than one data movement. This depends on the data groups it involves and all the actions it represents. In fact, in the case of OO-H, we observed that an I-Link can behave as Entry, Read and Exit, while each data group involved in the execution of an S-link represents a Write.

### *6.2. Application of the guidelines to WebML*

In order to show the applicability of the guidelines provided, we applied them so as to conceive the mapping and measurement rules for a COSMIC measurement procedure related to WebML [76], which is one of the best-known model-driven approaches as regards designing and developing Web applications. It is widely used and employed in both academia and the business context and specifies the structure of a Web application, the organization, and the presentation of its content in a hypertext. WebML also supports the use of the XML syntax to automatically realize the Web site. The specification of a Web application in WebML consists of three perspectives : a Structure model, which contains the data organization models ; a Hypertext model, which contains the navigation primitives like Unit, Link, and Page ; and a Presentation model, which contains graphical interface aspects (see Figure 5).

The application of the guidelines provided in the previous section allowed us to obtain the mapping and measurement rules shown in Table 10 and Table 11, respectively. In particular, with regard to identifying the FURs, Rule 1 in Table 9 suggests focusing on structure and navigational models, and it is therefore necessary to consider only the structure and the hypertext models of the WebML platform independent model (obtaining Rule 1 in Table 10). The whole Web application is considered as a single layer (Rule 2 in Table 10). With regard to identifying the scope, Rule 3 in Table 9 allows us to derive

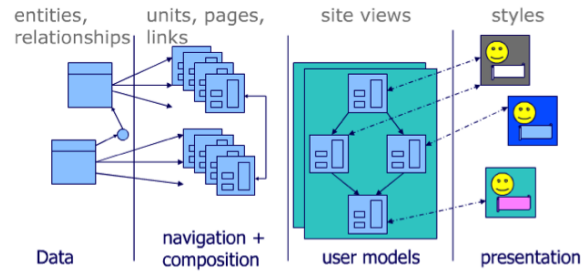


FIGURE 5: Organization of WebML models [76].

Rule 3 in Table 10, since we can consider the whole application or a part of it by correctly selecting the (subset of) Site Views from the Navigation model that represents the operations in which we are interested, and the connected Entities of the Structure model. Similarly to OO-H, in the case of WebML a functional user is each entity in the structure model that represents a User type (see Rule 4 in Table 10).

Rules 6 and 8 in Table 9 allow us to establish that, in the case of WebML, it is possible to identify a functional process as a single Site View composed of a set of Pages of which the Home page has an “H” symbol set, since this attribute represents the triggering event for the functional process (obtaining Rules 6 and 8 in Table 10). In fact, a Site View can be considered as a portion of the navigation model included in the measurement scope which represents a complete set of operations that perform a functionality, which is commenced with an event triggered by a Functional User. Units and Pages can consequently be exploited to identify functional sub-processes (see Rule 7 in Table 10).

Rules 9 and 10 in Table 9 allow us to draw the conclusion that in the case of WebML, the data groups can be identified from the entities of the Structure model, including aggregation, composition, and inheritance (thus obtaining Rules 9 and 10 in Table 10). In fact, data groups involve modeling primitives of the structure model because they represent the static part of the system and are considered to be the data (composed of atomic information, i.e., the attributes) managed by the application. Furthermore, a Page that appears in the Hypertext model can also be considered as a data group since it represents an informations container in the navigation model, i.e., a Web page of the Application.

The guidelines with which to identify measurement rules provided in

TABLE 10: COSMIC-based mapping rules for WebML

COSMIC concept	Rule	WebML modeling primitives
FURs	1	WebML platform independent model containing the structure and the hypertext models
Layer	2	The whole Web Application
Scope	3	The whole set of structure and the hypertext models OR a subset of Site Views and (related) subset of entities in the structure model
Functional User	4	Each entity in the structure model that represents a User type
(Inside the) Boundary	5	The structure and the navigation models within the established scope
Functional Process	6	A Site view
Functional Sub-Process	7.1	A Unit
	7.2	A Page
Triggering Event	8	The home page in the Site View identified via the H symbol
Data group	9.1	An entity of the structure model ; an aggregation hierarchy ; a composition hierarchy ; an inheritance hierarchy
	9.2	A page that appears in the hypertext model
Data attribute	10	An attribute of an entity in the Structure model

Table 9 allowed us to obtain the rules in Table 11, which can be exploited to determine COSMIC data movements with which to size applications developed with WebML.

According to these rules, a Data Unit, which represents the information originating from the structure model, can act as Entry, Read and Exit because it obtains the information from the persistent storage (Read) by means of a request from the user (Entry) to whom the information is then shown (Exit). An Index Unit, which represents a list of elements originating from the structure model, can act as Entry, Read and Exit, for the same reasons as those of the Data Unit. A Multi-choice Unit, which represents a set of elements originating from the structure model, also behaves in the same manner. An Entry Unit represents an empty module that the user must fill in to complete a certain operation and it acts as an Entry. Since these Units are included on a Page, this element represents an additional Read and Exit because it is necessary as regards displaying the information, and is thus moved by the application in order to accomplish the operation. A Login Unit represents the Web application authentication operation by means of checking the data inserted with those contained in the persistent storage ; it thus abstracts

TABLE 11: COSMIC-based measurement rules for WebML

Data Movement	Rule	OO-H modeling primitives
1 Entry(E)	11.2	1 Data Unit
	11.3	1 Index Unit
	11.4	1 Multi-choice unit
	11.5	1 Entry Unit
1 Exit(X)	12.1	Data Unit
	12.2	1 Index Unit
	12.3	1 Multi-choice unit
	12.4	1 Page
1 Read(R)	13.1	Data Unit
	13.2	1 Index Unit
	13.3	1 Multi-choice unit
	13.4	1 Page
	13.5	1 Login Unit
1 Write(W)	14.1	1 Create Unit
	14.2	1 Modify Unit
	14.3	1 Delete Unit

a Read data movement because the data stored must be retrieved from the persistent storage in order to perform the check. Finally, a Create Unit, a Modify Unit and a Delete Unit can perform a Write data movement because they modify the content of the persistent storage. It is worth recalling that in the case of these three Units there will be a data movement for each Data Group involved.

## 7. Related work

To the best of our knowledge, apart from the OO-HCFP measure proposed in [32] and detailed and assessed in the present paper, there is only one more recent work that provides a COSMIC measurement procedure with which to size model-driven Web applications. This was research by Ceke et al. [80], who proposed a procedure for estimating the size of Web applications developed using another model-driven approach named UWE [3]. This procedure includes mapping rules with which to estimate functional size from use cases and activity diagrams modeled with UWE. Unlike our proposal, no tools were provided for the automatic application of the procedure. Ceke and Milasinovic [13] recently performed an empirical study to assess the propo-



sal for effort estimation. They employed 19 Web applications designed with UWE by professional developers and applied the procedure proposed in [80] in order to obtain the functional size in terms of COSMIC. Similarly to our study, they considered simple linear regression to build the estimation models and leave-out-cross validation to validate the estimates obtained. Unlike our study, and despite their mention of several issues from the literature related to the use of MMRE, MdmRE, and Pred (25) [67]), only these summary measures were employed to assess the estimates obtained. Furthermore, we replicated the analysis performed by these authors and observed that they log transformed data, since the distributions considered did not satisfy the underlying linear regression assumptions. However, in the application of the estimation model obtained they did not transform back to their original scale of the variables, signifying that they compared the log transformed actual effort with the predicted effort obtained with the model using log transformed values. This can be considered as quite unfair in this kind of study. If we consider the effort estimation model transformed back to the raw scale, the values of MMRE, MdmRE, and Pred (25) are quite a lot worse than those reported in our paper.

Some model-driven measurement procedures have been devised based on the first generation of FSM to size Web applications [37, 38, 81]. Furthermore, other approaches have exploited design measures obtained from conceptual models specifying Web applications [39], approximate COSMIC size measures [82], and size measures for Web applications developed using a content management framework [83, 84]. Finally, there are other COSMIC measurement procedures for sizing other kinds of software, such as object-oriented systems [77, 78, 79] and real-time embedded systems [85]. We focus on the approaches for Web applications below.

In [37] a measurement procedure for Web applications developed using the model-driven method OOWS (Object-Oriented Web Solutions) was proposed. The method, denominated as OO-Method Function Points for the Web (OOmFPWeb), was designed to conform with the IFPUG counting rules for FPA. The efficacy and likely adoption of OOmFPWeb were evaluated using four datasets gathered from a family of experiments conducted in Spain, Argentina, and Austria [86]. The results of the study showed that OOmFPWeb is efficient, provides reproducible functional size assessments, and is perceived to be easy to use and useful by its users. However, the usefulness of OOmFPWeb for Web effort estimation has not yet been assessed.

Another model-driven approach with which to automatically size Web ap-

plications in terms of IFPUG FPA was provided in [81] for Web applications built using WebML and its supporting tool WebRatio [2]. An initial assessment of the approach was performed by comparing the size automatically obtained in terms of Function Points with the functional size computed by two skilled analysts who applied FPA manually. They used data that came from four separate projects developed by different companies using WebML, and the results showed that the sizes obtained automatically differed from those counted manually by a maximum amount of about 11%. However, further analysis with a larger dataset is required to generalize the results [81].

The OO-HFP measurement procedure with which to size Web applications developed with OO-H introduced in [38] is presented in Section 3.2. A first empirical assessment of the OO-HFP measure for the estimation of effort used a dataset of 12 Web applications to assess its effectiveness as regards effort estimation when compared to another set of measures, namely those defined by Mendes et al. [46] for the Tukutuku database. The results of the study revealed that the estimates obtained were comparable with those achieved using the Tukutuku measures. The approach was further assessed in [17] using the industrial Web projects employed in our study and by comparing its effectiveness as an effort predictor with that of IFPUG FPA. The results showed that the effort estimates obtained using OO-HFP were more accurate than those obtained using IFPUG FPA. We would like to reiterate that we have compared the accuracy of OO-HCFP with that of OO-HFP in this study, and the COSMIC-based approach performed better than the IFPUG FPA-based approach.

Baresi and Morasca [39] presented three empirical studies based on the use of W2000 [1], a special-purpose notation for the design of Web applications with the aim of investigating certain goals concerning the effort needed to design Web Applications using a model-driven approach. One of these goals concerned a metrics proposal with which to measure both the size and complexity of W2000 artifacts, and the consequent assessment of the impact of those metrics on the total effort needed to design Web applications. With regard to the size metrics, they proposed eleven metrics distributed among presentation, navigation, and content, which are the three main layers around which W2000 models are organized. The empirical results highlighted that different studies correlate different size measures with the actual effort, and do not therefore provide a clear indication of which of these measures are the most suitable as an effort predictor.

## 8. Conclusions and Future work

We have proposed a measurement procedure denominated as OO-HCFP for automatically sizing Web applications developed with a model-driven Web development method. The objective of the procedure proposed is to obtain a functional size based on COSMIC during the design phase of the development process by using only the conceptual models.

We have assessed OO-HCFP in the context of Web effort estimation by exploiting data from 30 Web applications developed by a Spanish company using the OO-H method and the VisualWADE tool. The results of the empirical analysis indicate that OO-HCFP measure can be considered as a suitable predictor of Web application development effort. In addition, the results show that the estimates achieved with OO-HCFP (i.e., a second generation of FSM methods) are better than those obtained with OO-HFP (i.e., a first generation of FSM methods) and OO-H design measures (i.e., IL and TL). In the following subsections we will discuss the implications [87] of our work for practitioners, researchers and educators, and future work.

### *8.1. Implications for Practitioners*

We believe that OO-HCFP is highly industry-relevant. The size and effort estimation of model-driven development projects poses new challenges to the project managers that should take into account the specific characteristics of Web applications and model-driven development. While a significant number of studies have been published recently in Web effort estimation, as far as we know, only one of them has focused on exploiting data provided by a Web company who follows a model-driven development approach. As reported by [5] factors such as better communication between stakeholders with models, quick response to changes due to automation and maturity of toolset can affect effort estimation in MDE. On the other hand, recent results in cost estimation for model-driven engineering (MDE) [7] has focused on explaining how COCOMO II cost drivers are likely to be influenced by development practices characteristics of MDE but they do not provide a effort estimation approach that can be used early on the development process. The novelties of our paper lays in providing a measurement procedure that helps project manager (and other team members without special training and certification in size estimation) in accurately estimate the size of a Web application developed following a model-driven development approach. The size measure is

obtained in an early phase of the Web development phase (conceptual modeling) and the results of our empirical analysis indicate that this measure can be considered a suitable predictor of Web application development effort. We believe that these results are of interest to those Web companies that need to improve their size and effort estimation processes. Note that other functional size measurement procedures have been proposed recently but their usefulness for Web effort estimation has not yet been assessed.

Moreover, it is important to note that OO-HCFP was automated in VisualWADE, signifying that a size measure of a Web application can easily be calculated when the OO-H conceptual model is specified. This avoids the ambiguity related to interpreting the COSMIC measurement rules and the need for special training to estimate functional size in an accurate and repeatable manner. The implications for an average IT person (e.g., project manager, developer) are an improvement in his/her productivity as regards estimating size and development effort, in addition to the accuracy of the estimates obtained.

The effort estimation models built for this particular company may also be useful for other similar companies who do not have historical data on effort estimation. Nevertheless, the results obtained should be further validated using larger datasets including data from other software companies [88, 89, 90, 91, 92, 93], and in the future we therefore plan to apply OO-HCFP in other contexts and also consider larger Web project datasets.

The expertise that a company should have to adopt our method includes basic knowledge on project management to apply the functional size measurement procedure, to use the effort estimation models and to monitor project life cycle activities (e.g., collecting metrics such as person/hours for modeling and testing models, effort for making manual changes, number of staff, team experience using the modeling tool and programming languages). Note that specific knowledge on functional size is not needed since the measurement procedure has been automated. However, companies adopting our method for other model-driven development approaches must perform the mapping between the concepts from the standard FSM method (i.e., COSMIC) and the modeling primitives of the selected model-driven method as described in Sections 4.1 and 4.2. For this purpose, the involvement of an expert or certified in COSMIC is recommended. In addition, the effort estimation models should be tailored to a particular organization's context meaning that the web company should collect historical data on past projects to calibrate their effort estimation models.

## 8.2. Implications for Researchers

Automated FSM procedures are valuable for use early in a new web projects life and/or can save time and effort compared with sizing accurately and consistently using the standard COSMIC measurement method. However, FSM procedures need to be properly designed by means of a mapping between the concepts from the standard FSM method and the modeling primitives of the selected model-driven development method. To provide more evidence about the mapping correctness, we plan to perform a conformity evaluation of OO-HCFP when compared to the ISO/IEC 19761 (COSMIC) standard. We are currently contacting COSMIC certified practitioners so as to perform such an evaluation by applying the process described in the ISO/IEC 14143-2 standard [94].

From a research perspective, the empirical study was a valuable means to provide evidence of the usefulness of a COSMIC measurement procedure for a model-driven Web development process. However, further studies using larger datasets of Web projects collected from culturally and organizationally different Web companies is needed to ensure the external validity.

Another interesting research direction might be the assessment of the contribution of the individual OO-HCFP components (i.e., Entry, Exit, Read, and Write) toward estimating the Web application development eort with regard to the total amount of OO-HCFP measure, as performed in other works (e.g., [95, 96, 97]). It is worth noting that OO-H can be considered as a method that is representative of the whole set of model-driven Web development approaches [18]. In the paper, we have attempted to convey the experience gained for the formulation and use of the proposed COSMIC measurement procedure by providing a set of guidelines that can support researchers/practitioners in the definition of a COSMIC measurement procedure concerned with the model-driven approach in which they are interested. We have shown the results of the application of those guidelines to the WebML approach, thus providing a type of preliminary evaluation of those guidelines. We plan to apply them to other approaches in the future.

OO-HCFP is based on fundamental principles of measurement theory since it is based on COSMIC. We have also contributed to the theory of software measurement by clarifying the relationship between software measurements at the conceptual model level, quantifiable properties of software, and objects of measurement for the specific case of Web applications developed with OO-H.

### 8.3. Implications for Educators

To facilitate teaching Web project measurement, materials are provided to support Web application functional size measurement : A guideline containing the proposed mapping and measurement rules and the application of OO-HCFP in a web project. These materials are available at <http://www.dsic.upv.es/~sabrahao/ist18>. In addition, we believe that the research process described in this paper as well as the derived guidelines (see Section 6) provide a step-by-step approach that can be easily applied to the definition of other variants of COSMIC or any other FSM measurement procedure for model-driven development processes.

### 8.4. Future Work

We plan to carry out an empirical study for assessing the effort involved when modeling with OO-H. As suggested by Jolak et al. [45], the creation of models consist of different cognitive activities : (i) designing, i.e., thinking about the design (ideation, key-design decision making), (ii) notation expression, i.e., expressing a design in a modeling notation and (iii) layouting, i.e., the spatial organization of model elements in a diagram. To better understand the effort needed for creating OO-H models we should run experiments to measure how much effort each of these cognitive activities takes.

We also plan to apply our mapping and measurement rules at a higher level of abstraction, i.e., on application mockups. Nowadays rapid-prototyping tools are used to represent though static mockups the initial requirements. Mockups are sketches, or depictive expressions of ideas [98]. This type of informal model supports the process of software design and helps designers to inspect and develop one design idea as well as reflect on other alternatives. Specifically, we will define a mapping between the mockups elements and the model primitives, in the same way we defined the mapping between the OO-H primitives and the code elements that COSMIC focuses on. The formal models (i.e., the ones defined with OO-H) would play the role of an intermediate layer that can facilitate the identification of the sensible concrete elements in the mockups (i.e., concrete page and navigation elements that are also visible in the mockups, and to define on them the metrics for size and effort estimation).

This will allow companies that do not follow a model-driven engineering process to have a size and effort estimation of their web application early on the process, thus extending the external validity of our approach. For the

companies that adopt model-driven engineering, this will also provide an alternative design strategy. Modelers may choose starting the modeling of their application using either an informal or formal model, and an OO-H model could then be automatically obtained from an informal model (mockups) by means of model transformations.

## References

- [1] L. Baresi, F. Garzotto, P. Paolini, From web sites to web applications: New issues for conceptual modeling, in : Conceptual Modeling for E-Business and the Web, ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling, Salt Lake City, Utah, USA, October 9-12, 2000, Proceedings, 2000, pp. 89–100. doi:10.1007/3-540-45394-6\_9.  
URL [http://dx.doi.org/10.1007/3-540-45394-6\\_9](http://dx.doi.org/10.1007/3-540-45394-6_9)
- [2] S. Ceri, P. Fraternali, A. Bongio, Web modeling language (webml): a modeling language for designing web sites, Computer Networks 33 (1-6) (2000) 137–157. doi:10.1016/S1389-1286(00)00040-2.  
URL [http://dx.doi.org/10.1016/S1389-1286\(00\)00040-2](http://dx.doi.org/10.1016/S1389-1286(00)00040-2)
- [3] N. Koch, A. Kraus, The expressive power of uml-based web engineering (2002).
- [4] J. Gómez, C. Cachero, O. Pastor, Conceptual modeling of device-independent web applications, IEEE MultiMedia 8 (2) (2001) 26–39. doi:10.1109/93.917969.  
URL <http://dx.doi.org/10.1109/93.917969>
- [5] J. Hutchinson, J. Whittle, M. Rouncefield, S. Kristoffersen, Empirical assessment of MDE in industry, in : Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011, Waikiki, Honolulu , HI, USA, May 21-28, 2011, 2011, pp. 471–480. doi:10.1145/1985793.1985858.  
URL <http://doi.acm.org/10.1145/1985793.1985858>
- [6] J. Hutchinson, J. Whittle, M. Rouncefield, Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure, Sci. Comput. Program. 89 (2014) 144–161.

doi:10.1016/j.scico.2013.03.017.

URL <http://dx.doi.org/10.1016/j.scico.2013.03.017>

- [7] S. Sunkle, V. Kulkarni, Cost estimation for model-driven engineering, in : Model Driven Engineering Languages and Systems - 15th International Conference, MODELS 2012, Innsbruck, Austria, September 30-October 5, 2012. Proceedings, 2012, pp. 659–675. doi:10.1007/978-3-642-33666-9\_42.  
URL [http://dx.doi.org/10.1007/978-3-642-33666-9\\_42](http://dx.doi.org/10.1007/978-3-642-33666-9_42)
- [8] M. Ruhe, D. R. Jeffery, I. Wiczorek, Using web objects for estimating software development effort for web applications, in : 9th IEEE International Software Metrics Symposium (METRICS 2003), 3-5 September 2003, Sydney, Australia, 2003, p. 30. doi:10.1109/METRIC.2003.1232453.  
URL <http://dx.doi.org/10.1109/METRIC.2003.1232453>
- [9] G. Costagliola, S. D. Martino, F. Ferrucci, C. Gravino, G. Tortora, G. Vitiello, A COSMIC-FFP approach to predict web application development effort, J. Web Eng. 5 (2) (2006) 93–120.  
URL <http://www.rintonpress.com/xjwe5/jwe-5-2/093-120.pdf>
- [10] S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, Using web objects for development effort estimation of web applications : A replicated study, in : Product-Focused Software Process Improvement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 186–201.
- [11] E. Mendes, Using knowledge elicitation to improve web effort estimation: Lessons from six industrial case studies, in : 34th International Conference on Software Engineering, ICSE 2012, June 2-9, 2012, Zurich, Switzerland, 2012, pp. 1112–1121. doi:10.1109/ICSE.2012.6227108.  
URL <http://dx.doi.org/10.1109/ICSE.2012.6227108>
- [12] G. Barabino, G. Concas, E. Corona, D. Grechi, M. Marchesi, D. Tiganò, Web framework points: an effort estimation methodology for web application development using a content management framework, Journal of Software : Evolution and Process 27 (9) (2015) 603–624. doi:10.1002/smr.1715.  
URL <http://dx.doi.org/10.1002/smr.1715>



- [13] D. Ceke, B. Milavinović, Early effort estimation in web application development, *Journal of Systems and Software* 103 (2015) 219 – 237. doi:<http://dx.doi.org/10.1016/j.jss.2015.02.006>.  
URL <http://www.sciencedirect.com/science/article/pii/S0164121215000394>
- [14] D. Azhar, E. Mendes, P. Riddle, A systematic review of web resource estimation, in : *Proceedings of the 8th International Conference on Predictive Models in Software Engineering, PROMISE '12*, ACM, New York, NY, USA, 2012, pp. 49–58. doi:[10.1145/2365324.2365332](https://doi.org/10.1145/2365324.2365332).  
URL <http://doi.acm.org/10.1145/2365324.2365332>
- [15] R. L. Glass, *Software Engineering : Facts and Fallacies*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [16] J. Gómez, C. Cachero, *Information modeling for internet applications*, IGI Global, Hershey, PA, USA, 2003, Ch. OO-H Method : Extending UML to Model Web Interfaces, pp. 144–173.  
URL <http://dl.acm.org/citation.cfm?id=949807.949816>
- [17] S. Abrahão, J. Gómez, E. Insfrán, Validating a size measure for effort estimation in model-driven web development, *Inf. Sci.* 180 (20) (2010) 3932–3954. doi:[10.1016/j.ins.2010.05.031](https://doi.org/10.1016/j.ins.2010.05.031).  
URL <http://dx.doi.org/10.1016/j.ins.2010.05.031>
- [18] N. Moreno, A. Vallecillo, Towards interoperable web engineering methods, *JASIST* 59 (7) (2008) 1073–1092. doi:[10.1002/asi.20811](https://doi.org/10.1002/asi.20811).  
URL <http://dx.doi.org/10.1002/asi.20811>
- [19] A. J. Albrecht, Measuring application development productivity, in : *Proceedings of the joint SHARE/GUIDE/IBM application development symposium*, Vol. 10, 1979, pp. 83–92.
- [20] Ç. Gencel, O. Demirörs, Functional size measurement revisited, *ACM Trans. Softw. Eng. Methodol.* 17 (3). doi:[10.1145/1363102.1363106](https://doi.org/10.1145/1363102.1363106).  
URL <http://doi.acm.org/10.1145/1363102.1363106>
- [21] International Standardization Organization (ISO), *ISO/IEC 14143-1 : 2003, Information Technology Software measurement - Functional Size Measurement. Part 1 : Definition of Concepts* (2003).

- [22] A. Abran, J. Desharnais, A. Lesterhuis, B. Londeix, R. Meli, P. Morris, S. Oligny, M. O'Neil, T. Rollo, G. Rule, L. Santillo, C. Symons, H. Toivonen, The COSMIC Functional Size Measurement Method Measurement Manual, version 4.0.1 In <http://www.cosmicon.com/portal/public/MMv4.0.1.pdf>. (2015).
- [23] B. Marín, O. Pastor, A. Abran, Towards an accurate functional size measurement procedure for conceptual models in an MDA environment, *Data Knowl. Eng.* 69 (5) (2010) 472–490. doi:10.1016/j.datak.2010.01.001.  
URL <http://dx.doi.org/10.1016/j.datak.2010.01.001>
- [24] H. v. Heeringen, E. v. Gorp, Measure the functional size of a mobile app : Using the cosmic functional size measurement method, in : 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, 2014, pp. 11–16. doi:10.1109/IWSM.Mensura.2014.8.
- [25] F. Ferrucci, C. Gravino, P. Salza, F. Sarro, Investigating functional and code size measures for mobile applications, in : 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 365–368. doi:10.1109/SEAA.2015.23.
- [26] F. Ferrucci, C. Gravino, P. Salza, F. Sarro, Investigating functional and code size measures for mobile applications : A replicated study, in : P. Abrahamsson, L. Corral, M. Oivo, B. Russo (Eds.), *Product-Focused Software Process Improvement*, Springer International Publishing, Cham, 2015, pp. 271–287.
- [27] T. Rollo, Sizing e-commerce, in : *Proceedings of Australian Conference on Software Measurement (ACOSM)*, 2000.
- [28] F. Ferrucci, C. Gravino, S. Di Martino, A case study using web objects and COSMIC for effort estimation of web applications, in : 34th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2008, September 3-5, 2008, Parma, Italy, 2008, pp. 441–448. doi:10.1109/SEAA.2008.60.  
URL <http://dx.doi.org/10.1109/SEAA.2008.60>

- [29] F. Ferrucci, C. Gravino, S. Di Martino, Estimating web application development effort using web-cobra and COSMIC: an empirical study, in : 35th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2009, Patras, Greece, August 27-29, 2009, Proceedings, 2009, pp. 306–312. doi:10.1109/SEAA.2009.47.  
URL <http://dx.doi.org/10.1109/SEAA.2009.47>
- [30] S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, Web effort estimation: Function point analysis vs. COSMIC, *Information & Software Technology* 72 (2016) 90–109. doi:10.1016/j.infsof.2015.12.001.  
URL <http://dx.doi.org/10.1016/j.infsof.2015.12.001>
- [31] L. De Marco, F. Ferrucci, C. Gravino, F. Sarro, S. Abrahão, J. Gómez, Functional versus design measures for model-driven web applications: a case study in the context of web effort estimation, in : Proceedings of the 3rd International Workshop on Emerging Trends in Software Metrics, WETSoM 2012, Zurich, Switzerland, June 3, 2012, 2012, pp. 21–27. doi:10.1109/WETSoM.2012.6226988.  
URL <http://dx.doi.org/10.1109/WETSoM.2012.6226988>
- [32] S. Abrahão, L. De Marco, F. Ferrucci, C. Gravino, F. Sarro, A cosmic measurement procedure for sizing web applications developed using the oo-h method, in : Proceedings of the Workshop on Advances in Functional Size Measurement and Effort Estimation, FSM '10, ACM, New York, NY, USA, 2010, pp. 2 :1–2 :8. doi:10.1145/1921705.1921707.  
URL <http://doi.acm.org/10.1145/1921705.1921707>
- [33] R. Wieringa, *Design science methodology for information systems and software engineering*, Springer, 2014, 10.1007/978-3-662-43839-8. doi:10.1007/978-3-662-43839-8.
- [34] J.-P. Jacquet, A. Abran, *Information technology software engineering software measurement - functional size measurement part3 : Verification of functional size measurement*. (1997).
- [35] ISO/IEC 14143-3. *From Software Metrics to Software Measurement Methods : A Process Model* (2002).
- [36] VISUALWADE, *Visualwade* <http://www.visualwade.com/> (2004).

- [37] S. M. Abrahão, O. Pastor, Measuring the functional size of web applications, *Int. J. Web Eng. Technol.* 1 (1) (2003) 5–16. doi:10.1504/IJWET.2003.003265.  
URL <http://dx.doi.org/10.1504/IJWET.2003.003265>
- [38] S. M. Abrahão, E. Mendes, J. Gómez, E. Insfrán, A model-driven measurement procedure for sizing web applications: Design, automation and validation, in : *Model Driven Engineering Languages and Systems, 10th International Conference, MoDELS 2007, Nashville, USA, September 30 - October 5, 2007, Proceedings, 2007*, pp. 467–481. doi:10.1007/978-3-540-75209-7\_32.  
URL [http://dx.doi.org/10.1007/978-3-540-75209-7\\_32](http://dx.doi.org/10.1007/978-3-540-75209-7_32)
- [39] L. Baresi, S. Morasca, Three empirical studies on estimating the design effort of web applications, *ACM Trans. Softw. Eng. Methodol.* 16 (4). doi:10.1145/1276933.1276936.  
URL <http://doi.acm.org/10.1145/1276933.1276936>
- [40] International Function Point Users Group (IFPUG), Hints to counting web sites. ifpug white paper. (1998).
- [41] B. Kitchenham, A procedure for analyzing unbalanced datasets, *IEEE Trans. Software Eng.* 24 (4) (1998) 278–301. doi:10.1109/32.677185.  
URL <http://dx.doi.org/10.1109/32.677185>
- [42] N. Condori-Fernández, O. Pastor, Verifying the construction of a software model from a requirements model, in : *Anais do WER06 - Workshop em Engenharia de Requisitos, Rio de Janeiro, RJ, Brasil, Julho 13-14, 2006, 2006*, pp. 138–145.
- [43] D. Meyerhoff, A. Golze, B. Laibarra, R. V. D. P. Kraan, A. Wallet (Eds.), *Software Quality And Software Testing In Internet Times*, Springer-Verlag, Berlin, Heidelberg, 2002.
- [44] D. Mills (Ed.), *Testing for Web applications - In Software Engineering for Modern Web applications : Methodologies and Technologies*, IGI Global, 2008.
- [45] R. Jolak, E. Umuhoza, T. Ho-Quang, M. R. V. Chaudron, M. Brambilla, *Dissecting Design Effort and Drawing Effort in UML Modeling*,

- in : 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2017, pp. 384–391. doi:10.1109/SEAA.2017.55.
- [46] E. Mendes, N. Mosley, S. Counsell, Investigating web size metrics for early web cost estimation, *Journal of Systems and Software* 77 (2) (2005) 157–172. doi:10.1016/j.jss.2004.08.034.  
URL <http://dx.doi.org/10.1016/j.jss.2004.08.034>
- [47] K. Srinivasan, D. Fisher, Machine learning approaches to estimating software development effort, *IEEE Transactions on Software Engineering* 21 (2) (1995) 126–137. doi:10.1109/32.345828.
- [48] L. C. Briand, I. Wieczorek, *Resource Estimation in Software Engineering*, John Wiley & Sons, Inc., 2002. doi:10.1002/0471028959.sof282.  
URL <http://dx.doi.org/10.1002/0471028959.sof282>
- [49] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Using evolutionary based approaches to estimate software development effort, in : M. Chris (Ed.), *Encyclopedia of Evolutionary Computation and Optimization Algorithms in Software Engineering : Applications and Techniques*, IGI Global, 2010, pp. 13–28.
- [50] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Genetic programming for effort estimation : An analysis of the impact of different fitness functions, in : *2nd International Symposium on Search Based Software Engineering*, 2010, pp. 89–98. doi:10.1109/SSBSE.2010.20.
- [51] F. Ferrucci, C. Gravino, F. Sarro, How multi-objective genetic programming is effective for software development effort estimation?, in : *Proceedings of the Third International Conference on Search Based Software Engineering, SSBSE'11*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 274–275.  
URL <http://dl.acm.org/citation.cfm?id=2042243.2042280>
- [52] F. Sarro, Search-based approaches for software development effort estimation, in : *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, Profes '11*, ACM, New York, NY, USA, 2011, pp. 38–43. doi:10.1145/2181101.2181111.  
URL <http://doi.acm.org/10.1145/2181101.2181111>

- [53] F. Ferrucci, M. Harman, F. Sarro, Search-based software project management, in : G. Ruhe, C. Wohlin (Eds.), *Software Project Management in a Changing World*, Springer Berlin Heidelberg, 2014, pp. 373–399. doi:10.1007/978-3-642-55035-5\_15.  
URL [https://doi.org/10.1007/978-3-642-55035-5\\_15](https://doi.org/10.1007/978-3-642-55035-5_15)
- [54] F. Sarro, F. Ferrucci, C. Gravino, Single and multi objective genetic programming for software development effort estimation, in : *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12*, ACM, New York, NY, USA, 2012, pp. 1221–1226. doi:10.1145/2245276.2231968.  
URL <http://doi.acm.org/10.1145/2245276.2231968>
- [55] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, E. Mendes, Investigating tabu search for web effort estimation, in : *2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2010, pp. 350–357. doi:10.1109/SEAA.2010.59.
- [56] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Using tabu search to estimate software development effort, in : A. Abran, R. Braungarten, R. R. Dumke, J. J. Cuadrado-Gallego, J. Brunekreef (Eds.), *Software Process and Product Measurement*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 307–320.
- [57] F. Sarro, A. Petrozziello, M. Harman, Multi-objective software effort estimation, in : *Proceedings of the 38th International Conference on Software Engineering, ICSE '16*, ACM, New York, NY, USA, 2016, pp. 619–630. doi:10.1145/2884781.2884830.  
URL <http://doi.acm.org/10.1145/2884781.2884830>
- [58] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, E. Mendes, How effective is tabu search to configure support vector regression for effort estimation?, in : *Proceedings of the 6th International Conference on Predictive Models in Software Engineering, PROMISE '10*, ACM, New York, NY, USA, 2010, pp. 4 :1–4 :10. doi:10.1145/1868328.1868335.  
URL <http://doi.acm.org/10.1145/1868328.1868335>
- [59] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, E. Mendes, Using tabu search to configure support vector regression for effort es-

- tion, *Empirical Software Engineering* 18 (3) (2013) 506–546. doi: 10.1007/s10664-011-9187-3.  
URL <https://doi.org/10.1007/s10664-011-9187-3>
- [60] E. Kocaguneli, T. Menzies, J. W. Keung, On the value of ensemble effort estimation, *IEEE Transactions on Software Engineering* 38 (6) (2012) 1403–1416. doi:10.1109/TSE.2011.111.
- [61] E. Mendes, B. A. Kitchenham, Further comparison of cross-company and within-company effort estimation models for web applications, in : 10th IEEE International Software Metrics Symposium (METRICS 2004), 11-17 September 2004, Chicago, IL, USA, 2004, pp. 348–357. doi:10.1109/METRIC.2004.1357920.  
URL <http://dx.doi.org/10.1109/METRIC.2004.1357920>
- [62] L. C. Briand, J. Wüst, Modeling development effort in object-oriented systems using design properties, *IEEE Trans. Software Eng.* 27 (11) (2001) 963–986. doi:10.1109/32.965338.  
URL <http://dx.doi.org/10.1109/32.965338>
- [63] M. J. Shepperd, S. G. MacDonell, Evaluating prediction systems in software project estimation, *Information & Software Technology* 54 (8) (2012) 820–827. doi:10.1016/j.infsof.2011.12.008.  
URL <http://dx.doi.org/10.1016/j.infsof.2011.12.008>
- [64] W. B. Langdon, J. J. Dolado, F. Sarro, M. Harman, Exact mean absolute error of baseline predictor, MARP0, *Information & Software Technology* 73 (2016) 16–18. doi:10.1016/j.infsof.2016.01.003.  
URL <https://doi.org/10.1016/j.infsof.2016.01.003>
- [65] F. Sarro, A. Petrozziello, Linear Programming as a Baseline for Software Effort Estimation, *ACM Transactions on Software Engineering and Methodology (TOSEM)*.
- [66] S. D. Conte, H. E. Dunsmore, V. Y. Shen, *Software Engineering Metrics and Models*, Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1986.
- [67] I. Myrtveit, E. Stensrud, Validity and reliability of evaluation procedures in comparative studies of effort prediction models, *Empirical Software*

- Engineering 17 (1-2) (2012) 23–33. doi:10.1007/s10664-011-9183-7.  
URL <http://dx.doi.org/10.1007/s10664-011-9183-7>
- [68] B. Kitchenham, L. Pickard, S. G. MacDonell, M. J. Shepperd, What accuracy statistics really measure, IEE Proceedings - Software 148 (3) (2001) 81–85. doi:10.1049/ip-sen:20010506.  
URL <http://dx.doi.org/10.1049/ip-sen:20010506>
- [69] J. P. Royston, An Extension of Shapiro and Wilk’s W Test for Normality to Large Samples, Journal of the Royal Statistical Society. Series C (Applied Statistics) 31 (2) (1982) 115–124.
- [70] V. Kampenes, T. Dyba, J. Hannay, I. Sjoberg, A systematic review of effect size in software engineering experiments, Information and Software Technology 4 (11-12) (2007) 1073–1086.
- [71] B. Kitchenham, L. Pickard, S. L. Pfleeger, Case studies for method and tool evaluation, IEEE Software 12 (4) (1995) 52–62. doi:10.1109/52.391832.  
URL <http://dx.doi.org/10.1109/52.391832>
- [72] B. Kitchenham, S. L. Pfleeger, L. Pickard, P. Jones, D. C. Hoaglin, K. E. Emam, J. Rosenberg, Preliminary guidelines for empirical research in software engineering, IEEE Trans. Software Eng. 28 (8) (2002) 721–734. doi:10.1109/TSE.2002.1027796.  
URL <https://doi.org/10.1109/TSE.2002.1027796>
- [73] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, Experimentation in Software Engineering, Springer, 2012. doi:10.1007/978-3-642-29044-2.  
URL <https://doi.org/10.1007/978-3-642-29044-2>
- [74] P. Runeson, M. Höst, A. Rainer, B. Regnell, Case Study Research in Software Engineering - Guidelines and Examples, Wiley, 2012.  
URL <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118104358.html>
- [75] M. Brambilla, Butti, Fifteen Years of Industrial Model-Driven Development in Software Front-Ends : from WebML to WebRatio and IFML, Available at <http://dbgroup.como.polimi.it/brambilla/sites/>



dbgroup.como.polimi.it.brambilla/files/referencematerials/ WebML-IFML-WebRatio-Novatica-EN.pdf (2015).

- [76] M. Brambilla, S. Comai, P. Fraternali, M. Matera, Designing web applications with webml and webratio, in : Web Engineering : Modelling and Implementing Web Applications, 2008, pp. 221–261.  
URL [http://dx.doi.org/10.1007/978-1-84628-923-1\\_9](http://dx.doi.org/10.1007/978-1-84628-923-1_9)
- [77] S. M. Abrahão, G. Poels, O. Pastor, A functional size measurement method for object-oriented conceptual schemas: design and evaluation issues, *Software and System Modeling* 5 (1) (2006) 48–71. doi:10.1007/s10270-005-0098-x.  
URL <http://dx.doi.org/10.1007/s10270-005-0098-x>
- [78] B. Marín, G. Giachetti, O. Pastor, Measurement of functional size in conceptual models: A survey of measurement procedures based on COSMIC, in : Software Process and Product Measurement, International Conferences : IWSM 2008, Metrikon 2008, and Mensura 2008, Munich, Germany, November 18-19, 2008. Proceedings, 2008, pp. 170–183. doi:10.1007/978-3-540-89403-2\_15.  
URL [http://dx.doi.org/10.1007/978-3-540-89403-2\\_15](http://dx.doi.org/10.1007/978-3-540-89403-2_15)
- [79] B. Marín, O. Pastor, G. Giachetti, Automating the measurement of functional size of conceptual models in an MDA environment, in : Product-Focused Software Process Improvement, 9th International Conference, PROFES 2008, Monte Porzio Catone, Italy, June 23-25, 2008, Proceedings, 2008, pp. 215–229. doi:10.1007/978-3-540-69566-0\_19.  
URL [http://dx.doi.org/10.1007/978-3-540-69566-0\\_19](http://dx.doi.org/10.1007/978-3-540-69566-0_19)
- [80] D. Ceke, M. Durek, S. Kasapović, Web application functional size estimation based on cosmic method and uwe approach, in : Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on, 2013, pp. 396–403.
- [81] P. Fraternali, M. Tisi, A. Bongio, Automating function point analysis with model driven development, in : Proceedings of the 2006 conference of the Centre for Advanced Studies on Collaborative Research, October 16-19, 2006, Toronto, Ontario, Canada, 2006, pp. 233–247. doi:10.1145/1188966.1188990.  
URL <http://doi.acm.org/10.1145/1188966.1188990>

- [82] L. De Marco, F. Ferrucci, C. Gravino, Approximate COSMIC size to early estimate web application development effort, in : 39th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2013, Santander, Spain, September 4-6, 2013, 2013, pp. 349–356. doi:10.1109/SEAA.2013.41.  
URL <http://dx.doi.org/10.1109/SEAA.2013.41>
- [83] E. Corona, M. Marchesi, G. Barabino, D. Grechi, L. Piccinno, Size estimation of web applications through web CMF object, in : Proceedings of the 3rd International Workshop on Emerging Trends in Software Metrics, WETSoM 2012, Zurich, Switzerland, June 3, 2012, 2012, pp. 14–20. doi:10.1109/WETSoM.2012.6226986.  
URL <http://dx.doi.org/10.1109/WETSoM.2012.6226986>
- [84] E. Corona, G. Concas, M. Marchesi, G. Barabino, D. Grechi, Effort estimation of web applications through web CMF objects, in : 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, Assisi, Italy, October 17-19, 2012, 2012, pp. 15–22. doi:10.1109/IWSM-MENSURA.2012.12.  
URL <http://dx.doi.org/10.1109/IWSM-MENSURA.2012.12>
- [85] H. Soubra, A. Abran, S. Stern, A. Ramdane-Cherif, Design of a functional size measurement procedure for real-time embedded software requirements expressed using the simulink model, in : 2011 Joint Conf of 21st Int'l Workshop on Software Measurement and the 6th Int'l Conference on Software Process and Product Measurement, IWSM/Mensura 2011, Nara, Japan, November 3-4, 2011, 2011, pp. 76–85. doi:10.1109/IWSM-MENSURA.2011.52.  
URL <http://dx.doi.org/10.1109/IWSM-MENSURA.2011.52>
- [86] S. Abrahão, G. Poels, A family of experiments to evaluate a functional size measurement procedure for web applications, *Journal of Systems and Software* 82 (2) (2009) 253–269. doi:10.1016/j.jss.2008.06.031.  
URL <http://dx.doi.org/10.1016/j.jss.2008.06.031>
- [87] R. Wieringa, M. Daneva, Six strategies for generalizing software engineering theories, *Science of Computer Programming* 101 (2015) 136 – 152, towards general theories of software engineering.

doi:<https://doi.org/10.1016/j.scico.2014.11.013>.  
URL <http://www.sciencedirect.com/science/article/pii/S0167642314005450>

- [88] F. Ferrucci, E. Mendes, F. Sarro, 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, ACM, New York, NY, USA, 2012, pp. 29–38. doi:10.1145/2365324.2365330.  
URL <http://doi.acm.org/10.1145/2365324.2365330>
- [89] B. Turhan, E. Mendes, A comparison of cross-versus single-company effort prediction models for web projects, in : 40th EUROMICRO Conference on Software Engineering and Advanced Applications, EUROMICRO-SEAA 2014, Verona, Italy, August 27-29, 2014, 2014, pp. 285–292. doi:10.1109/SEAA.2014.41.  
URL <http://dx.doi.org/10.1109/SEAA.2014.41>
- [90] E. Kocaguneli, T. Menzies, E. Mendes, Transfer learning in effort estimation, Empirical Software Engineering 20 (3) (2015) 813–843. doi:10.1007/s10664-014-9300-5.  
URL <http://dx.doi.org/10.1007/s10664-014-9300-5>
- [91] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, From function points to cosmic - a transfer learning approach for effort estimation, in : P. Abrahamsson, L. Corral, M. Oivo, B. Russo (Eds.), Product-Focused Software Process Improvement, Springer International Publishing, Cham, 2015, pp. 251–267.
- [92] L. L. Minku, F. Sarro, E. Mendes, F. Ferrucci, How to make best use of cross-company data for web effort estimation?, in : 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2015, Beijing, China, October 22-23, 2015, 2015, pp. 172–181. doi:10.1109/ESEM.2015.7321199.  
URL <http://dx.doi.org/10.1109/ESEM.2015.7321199>
- [93] E. Mendes, M. Kalinowski, D. Martins, F. Ferrucci, F. Sarro, 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, ACM,

New York, NY, USA, 2014, pp. 12 :1–12 :10. doi:10.1145/2601248.2601284.

URL <http://doi.acm.org/10.1145/2601248.2601284>

- [94] International Standardization Organization (ISO), ISO/IEC 14143-2 : 2003, Information Technology Software measurement -Functional Size Measurement. Part 2 : Conformity evaluation of software size measurement methods to ISO/IEC 14143-1 :1998”, International Organization for Standardization, Geneva, Switzerland, 2002 (2002).
- [95] L. Buglione, Ç. Gencel, Impact of base functional component types on software functional size based effort estimation, in : Product-Focused Software Process Improvement, 9th International Conference, PROFES 2008, Monte Porzio Catone, Italy, June 23-25, 2008, Proceedings, 2008, pp. 75–89.  
URL [http://dx.doi.org/10.1007/978-3-540-69566-0\\_9](http://dx.doi.org/10.1007/978-3-540-69566-0_9)
- [96] L. Buglione, F. Ferrucci, Ç. Gencel, C. Gravino, F. Sarro, Which cosmic base functional components are significant in estimating web application development? A Case Study, in : joined International Conferences on Software Measurement (IWSM/MetriKon/Mensura), Stuttgart, Germany, 2010, pp. 205–224.
- [97] S. Morasca, On the use of weighted sums in the definition of measures, in : Proceedings of the 2010 ICSE Workshop on Emerging Trends in Software Metrics, WETSoM 2010, Cape Town, South Africa, May 4, 2010, 2010, pp. 8–15. doi:10.1145/1809223.1809225.  
URL <http://doi.acm.org/10.1145/1809223.1809225>
- [98] B. Tversky, Visualizing thought, Topics in Cognitive Science 3 (3) (2011) 499–535. doi:10.1111/j.1756-8765.2010.01113.x.

## **AnnexeA. Examples of OO-H Conceptual Models**

Figures A.6, A.7, A.8, A.9, and A.10 show some examples of OO-H conceptual models obtained with the OO-H method.

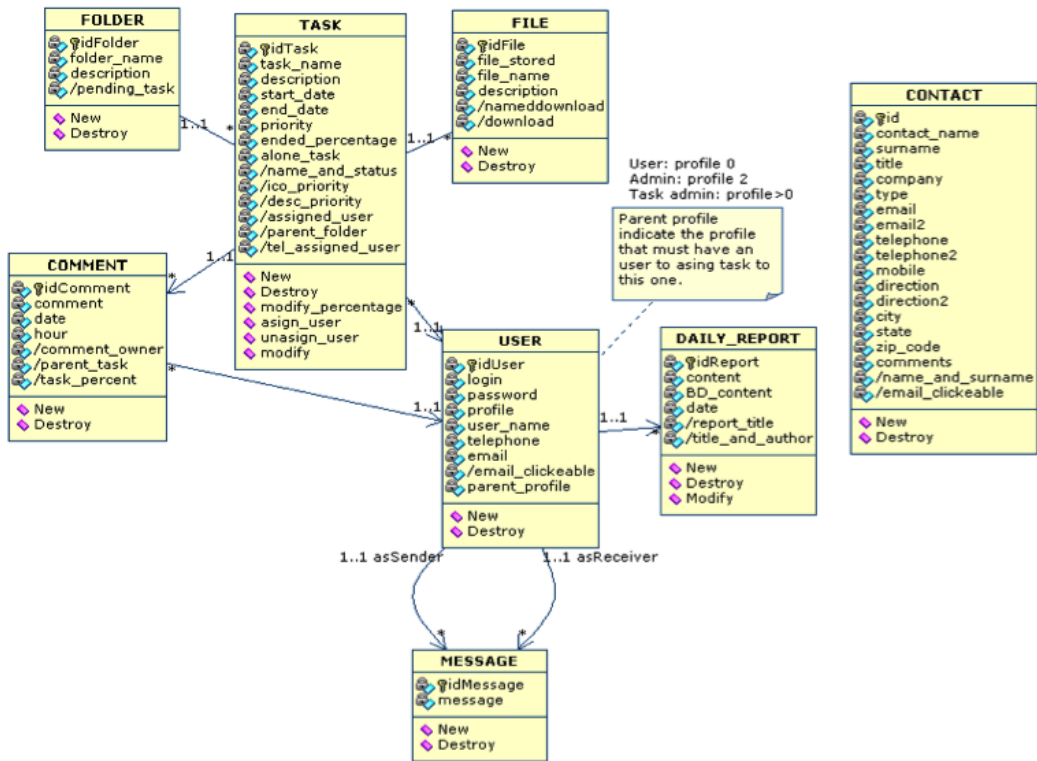


FIGURE A.6: The UCD for the Task Manager application [17].

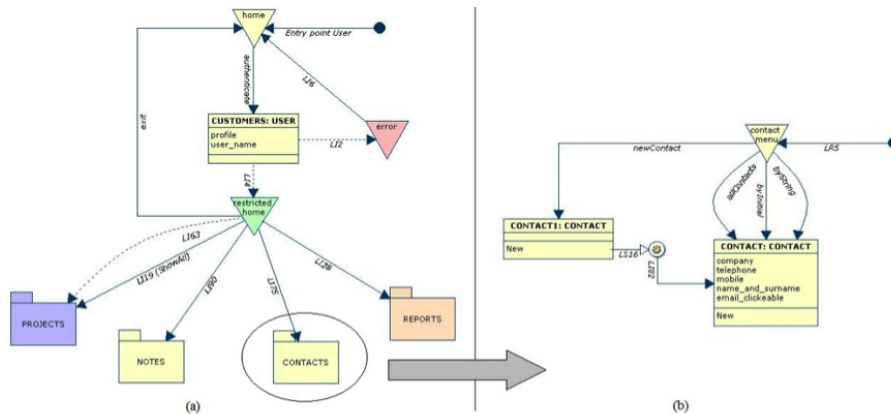


FIGURE A.7: NADs for the Task Manager application : (a) NAD level 0 for a User ; (b) Contacts Navigational Target [17].



