MMWA: a Software Sizing Model for Web Applications

Leonardo Mangia
Set-Lab, Software Engineering and Telemedia Laboratory
Dip.to Ingegneria dell’Innovazione - Università di Lecce
Via per Arnesano, 73100 Lecce, Italy
Leonardo.Mangia@unile.it

Roberto Paiano
Set-Lab, Software Engineering and Telemedia Laboratory
Dip.to Ingegneria dell’Innovazione - Università di Lecce
Via per Arnesano, 73100 Lecce, Italy
Roberto.Paiano@unile.it

Abstract

Estimating time and costs is a crucial factor in application development projects and low error margins are priority.

In line with the very fast evolution of Internet technologies, all applications are quickly becoming Web Applications, which are growing without a consolidated project methodology.

Thus there is a clear need for an estimation model for these applications' development projects.

The aim of the present paper is to illustrate a new Web Application cost estimation model that can form the starting point for any development project.

The estimation model described in this paper is called MMWA (Metrics Model for Web Applications). This is an "early measures model", since the measurements are effected at the start of the software life cycle, with a view to accurately estimate time and costs so that the right decisions can be taken concerning the development of the Web Application in hand.

1. Introduction and background

The fate of many organizations today depends on the accurate prediction of the time-to-market of a software product. Estimating the time is a crucial factor in an application development project.

Moreover, while the evolution of modern web technologies and of the applications that exploit these (Web Applications) makes it necessary to constantly update design methodologies, it also allows the modelling, not only of specific services (e.g. e-commerce, Internet Banking), but also key modules of a company's information system.

It is clear then that a tool for estimating the development time of a Web Application is indispensable if one intends to utilize modern web technologies for the realization of entire information systems or parts of these.

In this paper, section 1 briefly describes the context and the motivation of the work and defines a Web Application from the point of view of MMWA. Section 2 presents the MMWA model, emphasizing the requirements taken into account in the design and the general characteristics of the MMWA framework and the relative process of application.

Sections 3 and 4 present the two main sub-models of MMWA. These sections aim to give a brief idea of the composition of a sub-model in terms of components, tools and counting rules. The details, with all the references to the empirical values to be used, to the case studies to be considered, to the correct definition of the necessary information and application of the rules are the province of the MMWA model manual and are not dealt with in this paper. For reasons of simplicity the two optional sub-models are not described either.

Section 5, illustrates briefly the normalization and calibration phase, thus completing the description of the process of application MMWA. Section 6 presents the conclusions and discusses future developments.

1.1. Software sizing and Web Applications

Traditional estimation techniques [1][2][3][4][5][21], especially those aimed at the development of client/server systems, while conserving a significant applicability for projects linked to the Internet, are crucially different in certain respects.

Defining the differences between traditional applications and Web Applications means establishing guidelines to obtain new estimation models for the development of such applications which make as much use as possible of the know-how from previous estimation models. This is the key aspect of the approach used in designing MMWA.

In relation to the evolution of web technologies, today it is more exact to speak of hypermedia web-based applications or Web Applications, in which navigation
and the possibility of distributing information in such a way as to meet the needs of the users exist side-by-side with traditional operations such as transactions; operations whose execution can bring about changes to the informational and navigational structures of the Web Application themselves.

A Web Application then, in the context of MMWA and in accordance with concepts from the W2000 design model, is defined as the result of the integration of the hypertext paradigm with conventional applications. In this context, by conventional applications we mean those products which support traditional information systems and whose behaviour is typically transactional.

The complexity of a Web Application derives essentially from this convergence of software systems that were until recently quite separate.

It follows that a Web Application, a union of conventional and hypermedia applications, has four basic characteristics:
- Information Structures;
- Navigation;
- Operations;
- State Evolution.

The first two features have always been recognised in the literature as two aspects to be kept separate when tackling the design of a hypermedia application. In contrast, the third and fourth belong rather to those aspects which characterize conventional applications and are thus inherited from these.

The introduction of these operations to hypermedia applications has resulted in the evolution of their constituent elements. The information and navigational structures of the WAs do not simply remain as the designer originally specified, but evolve as a result of explicit operations by the user.

As far as sizing is concerned, it is necessary to quantify the level of dynamism supported by the WA. In order to remain consistent and simple the evaluation of the implementation effort for the elements that make up the WA takes account of the possible evolution of these.

These four aspects form the basis of W2000, a model for the design of Web Applications. The current version of MMWA uses certain elements of this model for the sizing of a Web Application. This is in line with the final objective of providing the designer with a complete framework (design model and support tools: W2000; sizing model: MMWA) for the management of Web Application projects. The four aspects mentioned above are accompanied by others (defined as secondary)
- Classes of users;
- Families of applications;
- Multi-device Delivery
- Customization;

- which all affect the complexity of a Web Application.

These aspects, like the previous ones, are considered in the W2000 design model and in MMWA.

2. A Metrics Model for Web Applications

There are few studies in the literature seeking to estimate the development effort for a Web Application [14][15][16][17][18][19] at the start of the software life cycle, and these studies are based on isolated ideas not linked to precise design models. Thus, at the present time, there is no well-defined metrics model able to measure the development time of a Web Application; nor is it possible to adapt the most well-known models (CocomoII [2][3] and Function Point Analysis [4][5]) to the domain of these software applications.

MMWA is a methodology that makes it possible to estimate the dimensions of a web application, evaluating those aspects that distinguish them from a conventional application, and which are easily neglected by an analysis carried out using traditional software sizing techniques. MMWA is based on W2000, a consolidated methodology for the design of Web Applications.

2.1. MMWA requirements

The requirements taken into consideration in the design of MMWA are:
- Independence of the development platform.
- Personalization of the model: the estimator must be able to insert, albeit in a guided way, the relevant aspects of their own operating context into the metrics.
- Critical error threshold of 20%.
- Project management support [6][7][8].
- Respect for the traditional phases of the software sizing process and possible automation: the model follows the flow of activity typical of a parametric approach, with an initial "rough" estimation phase based on the evaluation of certain parameters which constitute a metrics formula, and a subsequent phase in which the result obtained is calibrated on the basis of characteristics that do not depend on the application. The execution of the model must be schematic and rigorous so as to be automated by a metrics software tool. The achievement of this objective ensures that the model has practicality and speed of implementation.
- Provision of an estimate in actual work-hours.

The application of MMWA requires a preliminary but careful conceptual analysis of what is to be developed. The primary information source is made up of the results (codified in diagrams and matrices) of a process of formalization of the requirements. With reference to the cycle of development of a web application (fig. 1), the
moment in which the estimation process begins corresponds to the earliest phases of modelling (Start-Up).

Figure 1. MMWA in the cycle of development of a Web Application

In figure 1, the design in-the-large phase, independently of the design methodology adopted, should be understood as the phase in which the documents to be discussed with the customer are produced and in which the characteristics of the application are outlined, in order to take the first overall decisions concerning the design, or to validate those already taken.

In the elaboration of the model, it is assumed that the information on the characteristics of the WA gathered during the design in large phase is necessary for the evaluation of development times. The main aspects of design in-the-large are: Operations, State Evolution, Information Structures and Navigational Information Structures; secondary aspects are Classes of users, Families of applications, Multi-device fruition and Customisation.

Something that emerges clearly is the inapplicability of the metrics based on the source code lines. Indeed, an approach of this type would be wrong because, for this type of application (HTML, VBscript, etc…), the code is partly generated automatically by advanced authoring applications (Microsoft Frontpage, Microsoft Visual Interdev, Microsoft .Net, Macromedia Flash, etc…). Furthermore, even if one wished to apply an estimation model of this type, it would be an arduous task to identify the source code lines in a program written with a markup language, in which a number of instructions may be found on a single line.

The use of a functional metric, based on the analysis of the function point, would be reductive because, as is well known, with respect to a conventional application a Web Application has other fundamental aspects (navigation, publishing, content, interaction dynamics), apart from the functional side, which cannot be neglected because they noticeably affect development time.

MMWA metrics represent a solution to the problem of sizing a WA because they take all factors in the complexity of development of a WA into account.

2.2. The MMWA framework

Figure 2 illustrates the framework of the MMWA model.

MMWA is structured into four distinct and complementary sub-models, each one related to a particular factor of complexity in a Web Application:

- **Functional Sizing Model** (MMWA.fsm: estimates the functional dimension of a WA)
- **Navigational Structures Sizing Model** (MMWA.nssm: evaluates the complexity of realization of the navigational structures)
- **Publishing Sizing Model** (MMWA.psm: estimates the required effort for the design, implementation and maintenance due to publishing [11][13])
- **Multimedia Sizing Model** (MMWA.msm: estimates the effort required to develop the multimedia content of the application).

Only the first two sub-models are obligatory.
components: the set of information necessary for the correct implementation of the metrics.
- tools: the coding and formalization diagrams of this information.
- counting rules: the guidelines for the identification of the cost indicators and the technique, via formulas and correspondence tables, for obtaining the final measurement.

Specifically, the tools enable the estimator to codify, using diagrams, the information gathered, preparing it for the estimate of their relative complexity and expressing this as Unadjusted WCP.

It needs to be made clear however that the tools proposed by MMWA represent a technical support for the sizing of the functional component of the WA, but do not constitute a restriction on it. The tools proposed should thus be understood as guidelines for the estimator who remains, in any case, free to adopt alternative tools to achieve the same ends, appropriate to the specific design technique being applied.

Modularity is the main characteristic of MMWA. This modularity, besides providing operating flexibility, yields the following advantages:
- The possibility of carrying out corrective actions to one sub-model without affecting the internal structure of the others.
- The possibility of integrating and/or substituting further estimation sub-models.
- Greater support for project management activity.
- Greater adherence to diverse projects.
- Compatibility for an automation of the estimation process.

The notation adopted by MMWA is \texttt{<object.component>} following the notation from the object oriented paradigm. Thus for example the Functional Sizing Model sub-model is given as MMWA.fsm.tools, while the tools specific to this sub-model are given as MMWA.fsm.components.

### 3. Functional Sizing Model

The metric system of this sub-model satisfies the specifications of the ISO/IEC 14143-1:1998 [22] international standard, and is compatible with software sizing evaluation methodologies.

The procedure consists of identifying all the macro-functions required by the application, distinguishing the specific class of users for whom they are intended and the specific device which is used to deliver them. The rules for sizing follow those for the analysis of the function point although they need to be rectified in order to adapt them to the peculiarities of the WA.

The main measurement is based on the size of the data moving associated with each macro-function and a series of weightings which characterize the complexity of managing the various kinds of data (input data, output data, reading from an external/internal database, writing to an external/internal database). In line with this principle, the estimator must collect a series of information units (components) set out in the table below (table 1).

<table>
<thead>
<tr>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1: The macro-functions required by the user</td>
</tr>
<tr>
<td>C.2: The input data for each operation (DEI)</td>
</tr>
<tr>
<td>C.3: The output data for each operation (DEO)</td>
</tr>
<tr>
<td>C.4: The information entities which conceptually model the database</td>
</tr>
<tr>
<td>C.5: The application limit of each operation</td>
</tr>
<tr>
<td>C.6: The level of interaction which the various users of the Web Application have in each operation</td>
</tr>
<tr>
<td>C.7: The compatibility of each operation with the delivery devices supported by the Web Application</td>
</tr>
</tbody>
</table>

Table 1. Components for Functional Sizing Model (MMWA.fsm.components)

The procedure begins with an estimate of the complexity obtained from the analysis of the data moving of a macro-function. From this we obtain not only the size of the complexity determined by the management of the data moving for each macro-function, but also the diversity of access to the data according to the different classes of users and of the different delivery devices supported. In this way two indices of functional variance are obtained, one relative to the various types of users and the other to multi-delivering, integrated in a single index (IMU) which represents a percentage adjustment factor of increase/decrease of the previously obtained complexity.

There are two tools introduced in the metrics module: the first (Entity Diagram), derived from the HDM design methodology [9], makes it possible to have a complete picture of the information structure of the application and will also be useful for the other metrics modules (fig 3, top), the second (Functional Task Diagram) makes it possible to set out schematically the composition of the data moving for each macro-function (fig 3, bottom).

Figure 3 (bottom) shows the example of the “Shopping Bag Check Out” operation, which transforms the content of a shopping bag being ordered. In detail:
- The input data relating to the type of user (the non registered user must insert their name and address) and the type of device being used (no difference between PC and GPRS);
- For all users and all devices, the process acquires information from the internal database (ILD) relative to the price of the articles selected, any discounts, the codes of the articles and the cost of transport. For registered users only an additional discount is requested. When the user is connected via mobile
telephone no information is gathered technical (description and photo);

- In a database maintained by another application (ELD), it is assumed that client's level of reliability is extracted (as a result of a process external to the WA);
- The process stores the details of the order in the internal database (ILD);
- The confirmation of the order is given in the output, together with accompanying information.

The following observations complete the diagram:

- the "trigger" event represents the action which activated the process, in this example it might be a click on the 'order' button;
- a piece of information marked with an asterisk indicates that it is made up of more than one attribute, which the estimator leaves out in the diagram to maintain legibility but which are included in the subsequent counts;
- the internal database contains the WA data, while the external group of logical data is a pre-existing database.

The evaluation of the functional component will be guided by a set of 9 counting rules (MMWA.fsm.rules) which, by introducing a level of rigorosity in the metrics, make it possible to reduce the errors in applying the estimation technique.

The final formula for the estimate of this sub-model is shown in table 2. In this formula DM-tot is the estimate of the Data Moving total.

$$\text{SIZE}_{\text{MMWA.fsm}} = (1 + \text{IMU}) \times \text{DM}_{\text{tot}}$$

Table 2. Final Formula for Functional Sizing Model

![Table 2. Final Formula for Functional Sizing Model](image)

Figure 4 also shows an application of the sub-model to a real case.

4. Navigational Structures Sizing Model

The metrics proposed in the Navigation Structures Sizing Model measure the development effort for the navigational structures of a Web Application after a high-level modelling phase.

The level of information necessary for the Navigational Structures Sizing Model is such as to make these metrics applicable only after the first design phase of the Web Application, in which the characteristics of the application are defined in outline.

![Figure 4. A summary of the application of Functional Sizing Model](image)
For this reason, the information necessary for the estimator (MMWA.nssm.components) derive from a preliminary definition of the Web Application's information structures.

The measurement principle adopted by the Navigational Structures Sizing Model is based on the hypothesis that: the complexity of the navigational structures of a web-based hypermedia application increases in proportion to the density of the links which connect the informational objects that make up the hyperbase.

In this hypothesis it is assumed that the informational objects that make up the hyperbase of a Web Application have accessibility relations which enable the user to reach them via a direct navigational leap.

The higher the number of these relations, the more the user is able to “move” between elements of information content of a Web Application following different routes, and the greater the development complexity of the navigational structures.

The MMWA.nssm is based on graph theory, applied to the linked nature of a Web Application where the nodes are represented by information objects (source and destination of a navigational leap) while the arches represent the links which connect them.

The reference metrics were based only marginally on the theory of Paul De Bra and Geert-Jan Houben [16][17] which introduces the concepts of Adaptive Compactness and Adaptive Stratum to evaluate the quality of the navigational structure. Although this metric is significant, it is not directly adaptable to the context that we are analysing here because it is applied only in the post-implementation phase.

There is a close association between the current version of MMWA.nssm and the W2000 methodology [10][23]. It is important to note however that the adoption of a design methodology is necessary but not binding for the execution of MMWA.

The basic information on which the metrics (MMWA.nssm.components) will be developed is shown in table 3. In this table one type of node (C.1) defines a class of nodes [23]. A node is an "atomic" unit of information; that is, it represents a piece of informational content that cannot be further divided. The contents associated with a type of node are not new but are rather inherited from the type of informational structure (entity type, component type, semantic association type, collection type) from which they derive [10][23].

When we come then, to the modelling of the navigational structures, an explicit reference is no longer made to the informational structures of the hyperbase; rather, one speaks exclusively of navigational nodes.

An accessibility association (C.2) on the other hand, means determining which nodes can be reached by which other nodes. Finally, a cluster (C.3) is a set of node types and accessibility associations which link them [23]. All the node types in a cluster must derive from the same informational structure and every node type can be “shared” by different navigation clusters.

MMWA.nssm adopts the concept of clusters to evaluate the complexity of the navigational structures in that they are seen as a group of navigation clusters.

<table>
<thead>
<tr>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1: The types of nodes which constitute the navigational structure</td>
</tr>
<tr>
<td>C.2: The accessibility associations and the pattern of navigation between types of nodes</td>
</tr>
<tr>
<td>C.3: The navigation cluster</td>
</tr>
<tr>
<td>C.4: The visibility that the classes of users of the Web Application have of the navigational structure</td>
</tr>
<tr>
<td>C.5: The visibility that the delivery devices supported by the Web Application have of the navigational structure</td>
</tr>
</tbody>
</table>

Table 3. MMWA.nssm.components

The procedure avails itself of a single tool called the Cluster Diagram (fig 5), easily derived from the information structures of the Web Application, above all when one adopts a design methodology derived from HDM [9][10][12] (HDM2000, HDMlite, W2000).

The procedure avails itself of a single tool called the Cluster Diagram fig 5, easily derived from the information structures of the Web Application, above all when one adopts a design methodology derived from HDM [9][10][12] (HDM2000, HDMlite, W2000).
The first step is to evaluate the complexity of the navigation cluster that must be developed, on the basis of the number and the typology of nodes and the density of the links that these contain.

Subsequently the use of the so-called navigation patterns or navigation structures already defined (index-based navigation or guided tour, for example), which reduce the effort involved in developing this aspect of the Web Application, are also evaluated.

In this metrics module the increasing complexity resulting from the differing visibility which various user types or delivery devices have of the navigational structures is also evaluated. In this case we refer to indices of navigational variance.

Unlike the previous one, in this module the evaluation in UWCP of the navigation structures is based on a mix of the quantitative approach (adopted to describe the internal structure of the cluster) and the qualitative approach. With this approach UWCP values (navigation cluster complexity: WCI) are assigned on a 5-level scale of complexity: very low, low, medium, high, very high (figure 6).

The evaluation in this sub-model will be guided by a set of 5 counting rules (MMWA.nssm.rules).

The total size in UWCP of the navigational complexity of a cluster is determined using the formula shown in Table 4.

\[
(\text{CLUSTERSIZE})_{\text{UWCP}} = (1 + \text{IMU}) \times [(\text{WCI})_{\text{SL+IMU/A}}]
\]

Table 4. Final Formula for NSSM

In this formula:

- SL: number of Secondary Links identified in the cluster;
- IMU: percentage Increase index deriving from Multi-delivering and from the interaction of various classes of Users, evaluated using the appropriate table;
- A: factor of scale that takes into account the increasing complexity resulting from the presence of secondary links in the cluster. The factor of scale is equal to 2, the average of the results of a series of experimental observations, but can be modified by the estimator to enhance the reliability of the measurement. The range of admissible values goes from 1 to 3.

Figure 6 represents a summary of the application of the Navigational Structures Sizing Model (NSSM) for the entity "Paper" in a possible Web Conference Site Manager.

5. Normalization and calibration

Each sub-model is able to supply the measure of complexity required for the development of that particular component via a standard unit of measurement called UWCP (Unadjusted Web Complexity Point). This unit of measurement represents a scale of complexity and coincides with the positive axis of real numbers.

The individual contributions are applied to a normalization formula (table 5) which takes account of the different impact that each component has on the size of the Web Application. The final result is made up of the estimate of the difficulty of development of a web-based hypermedia application and is always expressed in WCP (removing the prefix ‘unadjusted’) (fig. 2).

\[
(\text{WEB APPLICATION SIZE})_{\text{WCP}} = C_0 (F\text{ Size}) + C_1 (N\text{ Size}) + C_2 (P\text{ Size}) + C_3 (M\text{ Size})
\]

Table 5. Normalization formula

In Table 5, F Size, N Size, P Size and M Size are the results of the separate sub-models, while the Ci coefficients represent the percentage impact of the i-th factor on the complexity of the WA. The MMWA framework proposes standard values for these coefficients, obtained from the average of the experimental results in various projects, but the estimator must be also able to evaluate these weightings in relation to the specifications of the project. Obviously depending on which of the two optional modules are applied, the
value of these weightings will vary in such a way as to conserve the same proportions (figure 2). The calibration phase takes into account factors concerning the characteristics of both the development environment of the project (the technologies adopted and the competence of the development team for example), and the type of application itself (figure 2). The effect of the calibration is the adjustment of the measurement of complexity, obtaining a value again expressed in WCP, in order to give the estimate a greater adherence to the specific project being analysed.

The rectification takes place via the evaluation of 11 cost indicators (figure 2), evaluated according to a scale of 6 values (from 0 to 5 in order of decreasing influence), specific to the project and having significant impact on its plan of development. The calibration procedure followed in MMWA is the same as that adopted in the metrics derived from Cocomo II [20] in that the weighting of these indicators affects the time of development exponentially; Table 6 shows the formula.

\[
(\text{Effort})_{\text{Work-Hours}} = A \times [(\text{Normalized Size})_{\text{WCP}}]^B
\]

Table 6. Calibration formula

In this formula:
- The constant A is used to capture the multiplying effects on effort of projects of growing dimensions and is called the linear increment constant. In consideration of the experimentation carried out up until now the value of this constant suggested in MMWA is 27.
- The scale factor B is the result of an algorithm which considers the influence of external characteristics on its time of development (which depends on the influence of the 11 cost indicators)

6. Conclusion and future work

The MMWA model has been used to weigh up Web Applications which are destined to provide innovative services for clients or which constitute new modules in the computer system of industrial and financial companies. This experience has made it possible to validate the approach and the architecture of MMWA; consider that the estimates obtained with MMWA have never been more than 12% \( (100 \times |\text{effective effort} - \text{estimate effort}| / \text{effective effort}) \) off the effort which the project actually required. This error margin is impressive when one considers that MMWA represents an early measure.

This experience has also demonstrated how easy the evaluation techniques in MMWA are to use; straightforward techniques based on information which emerges in the early planning phases of a WA. Indeed, even with projects whose estimates have been over 350 man days, MMWA's execution time, without the use of automation tools, has not exceeded 15 days. A further advantage that has emerged from this experience is that unlike early versions of CoCoMo [1], MMWA is independent of the logic of the programmer. This independence represents one of the principles of software sizing practice introduced by Allan Albrecht [21], the author of one of the most solid metrics regarding this criterion (Function Point Analysis), from which many others - and an international standard [22] - have been derived.

Considering that there is no reliable metric which has not been calibrated via intense experimental activity conducted in the field, the experience of using the MMWA metric will also continue on projects with different specifications and developed using different techniques of implementation and organization. This experience moreover, will make it possible to calibrate the empirical indicators present in MMWA with increasing accuracy.

Besides this intense experimental activity, there are two lines of future evolution for MMWA:
- definition and realization of an “MMWA Definition Model Manual” and a tool for the automation of metric procedures, with the objective of guiding, accelerating and reducing sizing process times;
- exploitation of the modularity of MMWA to define new sub-models or to modify existing ones in order to broaden the application target to include Web Applications not modelled with W2000, with the objective of generalizing MMWA and making it usable in any WA development context.

References


