Measuring the Functionality of Online Stores

Ernest Cachia and Mark Micallef

Software Engineering Process Improvement Research Group
Department of Computer Science and Artificial Intelligence
University of Malta
Malta

Abstract. This paper makes a case for the need of a framework which can be used to measure the functionality delivered by electronic commerce (e-commerce) systems. Such a framework would be helpful in areas such as cost prediction, effort estimatation, and so on. The paper also goes on to propose such a framework, based on the tried and tested methods of *function points* [1] and *object points* [5].

Keywords: Software Metrication, e-Commerce, Function Points, Object Points , Software Quality Assurance

1 Introduction

Measuring the size of a system before it is even built has been an important issue in computer science ever since non-trivial systems started being developed. When the size of a system is known, other metrics can be calculated based on that size. A few examples are *cost per unit*, *person months per unit* or even *errors per unit*. Having a repository of projects with such metrics would help in estimating how much a system could cost (for example) before it is even built.

Initial attempts at measuring system size resulted in metrics revolving around lines of code (LOC). However, such metrics had two major flaws [7] [8]:

1. System size could not be measured before it was actually built
2. LOC metrics for the same system built with different programming languages can vary wildly

In the late 70s, Albrecht [1] proposed function points as a means of measuring the size of a system (later refined and extended by Arthur [2]). Since then, new application domains have been opened which Albrecht would not have considered (or even been aware of) at the time. Such application domains may encompass systems such as real time systems, embedded systems, e-commerce systems and so on. It was the advent of such new types of systems that led researchers to extend or modify function points according to the characteristics of the systems in question. A few examples include Jones' *Feature Points* [3] for systems and engineering applications, Boeing's *3D Function Points* [4] for realtime systems, and Banker's *Object Points* [5] for business and database application software. Although these extensions do suffice for measuring the functionality of their target domains, the same cannot be said for measuring the functionality of e-commerce systems as outlined in [10].

Given the increase in the popularity of online shopping [9], it has become an almost discounted fact for businesses to offer their products and services online. With so much activity in this area,

this paper proposes that a framework utilising methods similar to *function points* [1] and *object points* [5] be developed so as to provide an easy and effective way of measuring the functionality of e-commerce systems.

With regards to terminology, of the many definitions given to e-commerce, one of the broadest and most complete definitions is the one coined by the British government as *"the exchange of information across electronic networks, at any stage in the supply chain, whether within an organisation, between businesses, between businesses and consumers, or between the public and private sectors, whether paid or unpaid"* [6]. Throughout this paper, references to e-Commerce should be taken to imply a Business-to-Consumer (B2C) type model. This decision is mostly based on limiting the scope of our work to manageable chunks for the time-being.

2 The need for better gauging features of E-Commerce Systems

When Albrecht [1] developed function points, he naturally reasoned about systems as being a collection of inputs, outputs, inquiries, files and external interfaces. This was true then, and still is now. However, with the advent of new paradigms and system types, this may not always be the ideal way to describe a system. In 1994, Banker [5] developed a new methodology that allowed its users to reason about a system at a higher level of abstraction than they would if they used function points. *Object points* defined a system as being a series of screens and reports which operated on a number of tables in a database. Naturally, this is not necessarily true for all types systems. However, systems to which this reasoning applied could suddenly be evaluated much quicker.

In 2004, Cachia [10] showed that e-commerce systems differed significantly from systems of other types. So much so that he managed to identify and rank a number of quality attributes which were of particularly high importance to e-commerce systems. According to Cachia, e-commerce systems differ from other systems mainly because they are:

- content-driven
- exposed and vulnerable to security risks
- accessed through WWW browsers, thus limiting a programmer's flexibility
- likely to have an enormous user base
- likely to change quite often

Based on these differences and a survey carried out amongst 350 online shoppers, Cachia identified the five most important quality attributes in e-commerce systems as being security, reliability, navigability, performance and portability.

Just as Banker [5] defined systems as being composed of a number of screens and reports, the authors of this paper are of the opinion that (based on Cachia's [10] findings) e-commerce systems too warrant a framework that uses terminology tailored to their nature. One could (for example) refer to various e-commerce components as being *online catalogues*, *shopping carts*, and so on. This makes it less tedious for stake holders to carry out scientific cost and effort predictions and also reduces their tendency to simply ad-hoc it.

3 Components of Online Stores

The proposed framework will consider B2C e-commerce environments to be made up of the following types of components:

– Static Web Pages (welcome page, contact information page, etc)
– Dynamic Web Pages (online catalogue, registration form, etc)

A static web page is, put simply, a web page that is written once and constantly served to users as is. No matter how many users view such pages over and over again, they will always see the same content, layout, colours, etc. Static web pages have the advantage of being very fast, reliable and easy to create. However, they can be time consuming to update and it can be easy to have broken links and associated problems because a web-page or link was missed when they were updated [11]. It is worth mentioning that it is very difficult to measure the functionality of static pages since they are, in effect, simply content with embedded links to other pages.

Dynamic web pages on the other hand, are generated by the server each time a user requests them. They are able to process a number of parameters passed to them by the user's browser and usually the resulting page is dependant on the nature of those parameters. Such capabilities have made it possible to show real-time information on websites, transform the WWW into an environment that can run applications and even offer the potential of personalisation. On the other hand, they are slow when compared to static web pages, require a much more powerful web server and are usually less reliable [11] since they contain program logic which may in turn contain errors. One must keep in mind that dynamic pages are usually supported by a number of backend services.

E-Commerce systems can also be said to be made up of a collection of objects of a higher abstraction. For example, one might examine a system and deduce that it is made up of an online catalogue, a product search page, an online ordering module and an online payment processing module. The proposed framework can also be used at this level of abstraction (see section 7) but at a core level, it will analyse static and dynamic pages. The reasoning behind this is that all the higher-level entities (online catalogue, product search page, etc) are in fact built out of pages.

4 Framework Overview

The proposed framework combines the ideas inherent in function point methodology [1][2] with Banker's [5] idea of using terms of a higher abstraction than inputs, outputs, interfaces, etc when measuring a system's functionality. Banker's work also serves as a good starting point because it deals with business and database applications, a genre of applications which shares some characteristics with e-commerce systems. However, instead of using Banker's terminology of screens and reports, the proposed framework will use such terms as static pages, dynamic pages, online catalogues, online ordering systems, and so on.

Calculating the size of the functionality of a system will be done in two steps. Firstly, a *page-by-page* analysis will be carried out. This will produce a so-called *unadjusted count* which is basically the sum of the points accrued by all individual pages in the system. However, this count must be adjusted according to certain criteria as discussed in section 6 below. This is necessary due to the fact that there may be certain system-wide requirements which are not countable at page-level but do in fact have an influence over the system-wide count.

5 The Unadjusted Count

During this phase, each page is individually analysed and assigned a count (or a number of points) according to the criteria defined below. The final unadjusted count is simply the summation of the counts of each individual page.

$$unadj\_count(system) = \sum_{i=1}^{n} unadj\_count(page_i)$$

5.1 Analysing Static Pages

Each static page is simply assigned a count of 1. This is because a static page is considered to be the simplest element in an e-commerce system which consists simply of content.

$$unadj\_count(static\_page) = 1$$

5.2 Analysing Dynamic Pages

Dynamic pages are assigned a count according to a number of different criteria which based on Cachia's [10] findings, will affect the effort required to develop each individual page. Firstly, for each page, one needs to know **the number of tables used** or manipulated by that page. This reasoning is similar to the reasoning employed by Banker [5] and are also seen to e-commerce systems due to the fact that they are largely content-driven [10].

The second evaluated criteria requires the user to estimate/predict the **expected popularity** of the page in question. Some pages within the same site will be visited repeatedly by multiple users (e.g. a product search page) whilst others may not (e.g. a testimonial submission page). If a page is expected to be popular, more care must be taken to make the page's internal workings efficient and able to cope with the repeated access, thus requiring more effort to develop.

If a page makes extensive use of **dynamic features** such as JavaScript^{TM}, VBScript ^{TM} and so on, it would in effect be providing more functionality to the user. This is also reflected in the testing effort in terms of compatibility testing across browsers, operating systems and devices. [12][10]. Hence, the level of use of dynamic features within a page is also considered an important criteria in our analysis.

With the advent of powerful web and database servers, **personalisation** has become feasible and research in this field is very active. Levels of personalisation range from a page displaying the message *"Hello Peter,"* when you log in, to making personalised product recommendations, to changing the whole structure and content of pages to suite your user profile. It is the opinion of the authoers that the unadjusted count of personlised pages should be higher than those which are not, due to the fact that they offer more functionality.

Finally, we take the page's visible **algorithmic complexity** into account. In some cases, what may seem like a simple form to the normal user may sometimes require vast amounts of processing in order to produce the required results. Hence, it is proposed that the unadjusted count for pages with a higher algorithmic complexity to be increased accordingly.

5.3 Points assigned to each criteria

The criteria defined above will be assigned points according to the following guidelines. The *"amount of tables"* criteria will be evaluated according to the table below:

# of Tables :	n=0	n ≤ 2	2 < n ≤ 5	5 < n
Points:	1	3	4	5

The remaining criteria are to be assigned a value of *low*, *medium* or *high* by the person carrying out the calculations, and the unadjusted count incremented accordingly. If one considers an individual criteria as being non-existent, no increment should occur for that particular criteria.

Criteria	None	Low	Med	High
Expected popularity of page	0	0	2	4
Dynamic Features	0	1	2	3
Personalisation	0	0	1	2
Algorithmic Complexity	0	1	2	4

6 Calculating the Adjusted Count

The unadjusted count needs to be modified due to reasons explained in section 4 above. One can calculate the adjusted count by answering a number of questions about the presence of system-wide characteristics. Each question is assigned a relevance value between 0 (not important or applicable) and 5 (absolutely essential). At first glance, this is very similar to Arthur's *complexity adjustment values* [2], however this is only true to a limited extent.

Firstly, following analysis of Arthur's questions they were deemed to be inapplicable in the context of e-commerce systems. Also, Arthur's formula can at most increase the function points of a system by 35% and the adjustments only increase the overall system count, never decrease. In some circumstances, the unadjusted count may need to be reduced. Finally, Arthur [2] assigns an equal weighting to the answer of each question whereas the proposed framework assigns different importance to different questions. For example, security considerations are considered to be more important than checking for dead links. Due to the very nature of e-commerce systems [10], the proposed framework poses different questions to those of Arthur [2] and also uses a different adjustment formula. The questions are as follows:

#	Question	Max Adj.
1	Does the system use and/or manipulate complex data?	5%
2	Does the site in question require extensive security measures?	10%
3	Are components with the system being build in such a way as to allow them to be reused in other systems in future?	5%
4	Is the system meant to provide multilingual features?	5%
5	Is it important that the system produced personalised output and behaviour?	5%
6	Is the site expected to be popular and receive vast amounts of daily visitors?	5%
7	Are feature changes to the site planned after its release?	5%
8	Is it important that the site be portable across browsers, operating systems and devices?	5%
9	Is the site to contain large numbers of links to pages in other sites?	2%
10	Does the system make use of previously built reusable components?	5%

Once the questions have been answered, the adjusted count can be calculated using the following formula...

$$AdjustedCount = UnadjCount \times adj\_factor$$

where:

$$adj\_factor = 100\% + \sum_{i=1}^{9} \left(\frac{A_i}{5} \times MaxAdj(Q_i) \right) - \left(\frac{A_{10}}{5} \times MaxAdj(Q_{10}) \right)$$

and:

$$A_i = Answer\ to\ question\ i$$
$$MaxAdj(Q_i) = Maximum\ adjustment\ of\ question\ i$$

Note that question 10 actually reduces the overall count because building the system out of reusable components would actually result in less effort being needed to develop it.

7 Quick Analysis

In some cases, it is simply not convenient (and may even be counter productive) to perform a detailed page-by-page analysis of a system. For this reason, the proposed framework also provides a way to perform a quicker (albeit less accurate) calculation. This is achieved by defining a number of higher level entities which one could analyse. For example, one could analyse a system as having 10 static content pages, an online catalogue, a product search page, and an online ordering page. The proposed framework provides a number of profiles whereby each high-level entity is pre-assigned the values for most criteria. So in the case of an online catalogue for example, the profile has a predefined popularity rating of high, a dynamic features rating of low, a personalisation rating of none, and an algorithmic complexity of none. With this information implemented in a software solution, estimates about a system's size could be calculated very quickly. This can come in very handy when clients ask for a quick cost estimate of a system during meetings. These profiles will be published on the web shortly and a software solution is also under development.

8 Conclusions and future work

The proposed framework enables stake-holders in e-commerce systems to be able to make informed decisions about the future of a project. Its simplicity is such that it reduces motivation to use ad-hoc methods and can be seamlessly integrated with any software development process. The proposed framework can be applied to development effort prediction, cost prediction and even the prediction of testing efforts and bug-counts, much in the same way as function points.

However, the weights and values assigned to each criteria should be verified through extensive case-studies and industry interviews so as to refine them and reinforce their validity. Once this has been done, the proposed framework will form an integral part of a larger project which aims to provide a number of metrics, tools and methodologies for measuring the quality of e-commerce systems.

References

1. Albrecht A.J., "Measuring Application Development Productivity", Proceedings of IBM Application Development Symposium, 1979
2. Arthur, L.J., Measuring Programmer Productivity and Software Quality, Wiley-Interscience, 1985
3. Jones C., "Applied Software Measurement", McGraw-Hill, 1991
4. Whitmire S.A., "An Introduction to 3D Function Points", Software Development, 1995
5. Banker, R. D. et. al., "Automating Output Size and Reuse Metrics in a Repository-Based Computer Aided Software Engineering (CASE) Environment", IEEE Transactions on Software Engineering, Vol. 20, No. 3, 1994
6. Prime Minister's Strategy Office, "E-Commerce@its.best.uk", www.number-10.gov.uk/su/ecomm/ec_body.pdf, 1999
7. Armour P. G., "Beware of Counting LOC", Communications of the ACM - Vol. 47 No. 3, March 2004
8. Kan S. H., "Metrics and Models in Software Quality Assurance", Addisson Wesley Professional, 2002
9. Emarketer.com, "Online Retail Update: Latest Q4 Quote", www.emarketer.com/news/article.php?1002631, Emarketer.com, 2004
10. Cachia E.,Micallef M., "Towards Appraising Online Stores", Proceedings of Software Engineering Knowledge Engineering (SEKE) Conference, 2004
11. Hean D. "Types of Web Publishing", http://www.yedit.com/web-content-management-system/314-types-of-web-publishing.html
12. Chandra K., Chandra S.S. "A Comparison VBScript, JavaScript and JScript", Journal of Computing in Small Colleges, 2003