www.ietdl.org

Published in IET Software
Received on 21st February 2013
Revised on 29th September 2013
Accepted on 15th October 2013
doi: 10.1049/iet-sen.2013.0031

ISSN 1751-8806

Role of unified modelling language in software development in Greece – results from an exploratory study

Panos Fitsilis, Vassilis C. Gerogiannis, Leonidas Anthopoulos

Business Administration Department, Technological Educational Institute of Thessaly, 41110 Larissa, Greece
E-mail: fitsilis@teilar.gr

Abstract: The Unified Modelling Language (UML) is a language for specifying, visualising, constructing and documenting software systems. The UML proved to be extremely successful and it has achieved tremendous popularity making it the de facto industry standard for object oriented system development. As such, many researchers presented empirical studies on the practical usage of UML but as well criticisms for UML complexity, difficulty to be learnt, etc. Even though a large number of articles and books are devoted to various aspects of UML language, there is little evidence on how UML is used. This study attempts to identify the profile of persons using UML, to pinpoint UML diagrams that are being used and their effectiveness, to discover whether CASE tools are being used and to record the perceived usefulness of UML language. For conducting the study a survey was developed and it was distributed to mailing lists of Greek IT professionals and to university students. The findings indicate that UML is used successfully in the majority of software development projects and that most users perceive UML positively since it supports faster system building, development of higher quality software systems, and for specific cases, it leads to software development cost-decrease.

1 Introduction

The Unified Modelling Language (UML) was developed as a unification effort of different object oriented development methodologies originally proposed by Booch [1], Jacobson *et al.* [2] and Rumbaugh *et al.* [3]. Immediately after its introduction, in 1997, UML was adopted by object management group (www.omg.org) and it became a software development standard. Several minor revisions (UML 1.3, 1.4 and 1.5) fixed shortcomings and bugs with the first version of UML, followed by the UML 2.0 major revision that was adopted by the OMG in 2005. The UML 2.0 specification is composed of UML infrastructure that defines the foundational language constructs required and UML superstructure, which defines the user level constructs required. To date, the latest version of UML is 2.4.1 and it is available through OMG web site [4].

The UML is a language for specifying, visualising, constructing and documenting software systems. The UML notation includes diagrams that provide different perspectives of the system under analysis or development [5]. The UML diagrams represent two different views of a system model: the static view, which defines the static structure of the system; and the dynamic (or behavioural) view that elaborates the dynamic behaviour of the system by showing collaborations among objects and changes to the internal states of objects. Even though, UML is a general-purpose modelling language however its usage is not only restricted to modelling software, but it is commonly used for system engineering, for business process modelling and for representing the organisational structures.

The UML achieved tremendous popularity making it de facto industry standard for object oriented system development [6]. As such, many researchers presented works, case studies and empirical studies on UML application, but as well criticisms for UML complexity, ambiguity, difficulty to be learnt etc. [7]. Even though a large number of articles and books are devoted to various aspects of UML language, there is little evidence on how UML is used. According to Grossman *et al.* [8], ‘there is a critical need for further research in all areas relating to UML usage. Very few empirical studies have been performed to date’. This is the exact topic that this paper is attempting to address. More specifically, this study attempts (a) to identify the profile of persons using UML, (b) to categorise the projects where UML is being used, (c) to pinpoint the most used UML diagrams and their effectiveness, (d) to discover whether CASE tools are being used and (e) to record the perceived usefulness of UML. The nature of this study is a statistical survey based on a questionnaire that was distributed electronically to the Greek software development community and the results obtained can be used in order to understand how today software systems are developed. According to the study of Kostoglou *et al.* [9], the majority of Greek companies (83%) employs up to 50 individuals, with only 10% of them to be considered as large enterprises with more than

100 employees, whereas their geographical range of activities is mainly national. The main activities of these companies vary considerably from service delivery to training, while information system development is considered as the third most important activity.

The rest of this article is structured as follows. Section 2 presents the related works and surveys on UML usage. Section 3 presents research questions (RQs) and research methodology. In Section 4, we present the results obtained on the RQs and finally in Section 5 we present the conclusions of this study.

2 Related work

Budgen *et al.* [10] in their study attempted to conduct a systematic literature review, aiming to identify empirical studies and publications on UML: model quality, metrics, comprehension, methods, tools and UML adoption. Up to the end of 2008, they have identified 49 relevant publications, and reported that 'researchers tend to use UML as a 'given' and seem reluctant to ask questions that might help to make it more effective'. He concluded that 'there are few studies of adoption and use in the field. As so many studies we have identified are laboratory-based, these do tend to be centred on one viewpoint, as well as relatively 'shallow' in the way that UML has been used'.

Batra [11] in two special issues of the Journal of Database Management studied the 'The past, the problems and the prospects' and 'Cognitive issues in UML Research' [12]. In the preface of this special issue [12], Batra presented a number of RQs related with UML usage: 'Are the UML constructs used formally with the support of CASE tools? Is the sequence diagram used to bridge the use case with the class diagram, or is it used to clarify the coding logic to the developer? Is UML primarily a way of communicating (i.e. informal), or a means to design software (i.e. formal), or both? Does it provide cognitive support to developers? Does it aid comprehension? Does it reduce cognitive load?'.

Dobing and Parsons [13] studied the nature of UML usage with a survey that was promoted by OMG. In their study, 182 respondents have participated. Among these 182 respondents, 171 responses came from UML users, whereas 11 came from UML users that were combining the UML with other languages and methodologies. Among the findings of this study, it is worth stressing that (a) the UML users mostly use class diagrams, use case diagrams and sequence diagrams, (b) use cases narratives, activity diagrams and use case diagrams are the preferred means with regard to customer involvement and (c) class diagram, sequence diagram and state machine diagram are considered as the most useful for capturing technical aspects. Further, the study revealed the main reasons for not using some UML diagrams. For example:

- About 50% of the respondents said that class diagrams and 48% of the respondents said that activity diagrams were not understood by analysts.
- Use case diagrams have insufficient value to justify development cost and their usefulness for developers is limited (42%).
- Collaboration diagrams capture redundant information (49%).
- State machine diagrams are not useful for most projects (42%).

In a later study on the same topic, Dobing and Parsons [14] attempted to identify the differences in UML diagram usage on software projects including frequency of diagram usage, usage purposes and the roles of clients/users in their creation and approval. In this survey, respondents did not report particular usage problems regarding user-friendliness of UML; they were more concerned with missing modelling features of UML such as support for user interfaces, security and database design etc.

Lang and Fitzgerald [15] studied the use and the adoption rates for UML constructs for web development and reported that the usage of use case diagrams/scenarios in their development projects is 72%, class diagrams is 62% and state diagram is 50%. Even though the reported usage is quite high, other techniques such as screen prototypes (97%), flowcharts (95%), storyboards (85%) and entity-relationship diagrams (74%) have proved to be more popular.

Grossman *et al.* [8] studied the adoption and usage of UML using the task-technology fit (TTF) framework [16]. TTF links performance with the fit between the task being performed and the particular type of technology being utilised. In their study, three main RQs were addressed: (a) Do individuals who use the UML perceive it to be beneficial? (b) Does UML provide a TTF to individuals who utilise it? (c) What are the characteristics that affect the UML use? In this study, a sample size of 131 UML users was obtained and the analysis revealed that the users surveyed had a positive perception of UML since they viewed the UML as accurate, consistent and flexible enough to use on development projects.

Industrial case studies and surveys give empirical evidence that individuals use UML in many different ways (even within the same project team). In the study of Lange *et al.* [17], participated 80 software architects, coming from different application domains, and having a different job profile, it was revealed that participants mostly use UML to define the 'use case view' and the 'logical view' of the system under development (according to Philippe Kruchten '4 + 1 view model' [18]) and that their adherence to the UML specification is rather loose. In the same study, it was suggested that UML practices should improve in areas such as: modelling uniformity, modelling standards, development of project specific reference architectures and patterns etc.

Wrycza and Marcinkowski [19] in a student questionnaire where 180 students participated proved that the complexity of teaching and using UML is high since more than 90% of the students have a preference for a lighter version of UML 2.0. The students assessed the user-friendliness of the diagrams, finding use case and class diagrams as the most users' friendly, whereas specialised diagrams like 'interaction overview' and 'composite structure' were considered as the less appealing. Four diagrams were selected as the diagram of a light version: use case, class, activity and sequence diagrams. It seems that for most projects, a 'light' or limited version of UML will suffice.

Similarly, at the national level, Peneva *et al.* [20] investigated the utilisation of UML in Bulgarian small and medium companies as a modelling language that facilitates the construction of traceable models. The goal was to evaluate the current practices as well as the needs for UML training. Their study concluded that in most cases UML is not properly used in the software industry and more UML training is needed with more emphasis on methodological issues.

doi: 10.1049/iet-sen.2013.0031

A different approach was used by Gu *et al.* [21] that proposed a model based on IT adoption framework and organisational culture theory. This model identifies fourteen variables, covering seven categories (IT characteristics, organisation technology, environment, organisation structure, organisation process, organisation culture and project culture) that influence UML adoption in organisations. The results of the study among others indicate that larger companies adopt UML easier, that UML adoption is influenced by stronger and results oriented leadership and that better team communication and collaborative team work of in project culture, influence UML adoption.

A summary table of the main works on UML usage is presented in Table 1.

The aim for conducting this research stems from the fact that no studies were developed during the past years on how UML is used, neither at international nor at national level. This type of study will offer a valuable insight on the most used UML diagrams, on the type of projects where UML is used etc. knowledge that can be used for UML curriculum development and for more focused UML training.

3 Research methodology

The approach that is undertaken for this study comprises two components. The first component is a literature review (Section 2), and an exploratory survey. The literature review is necessary in order to establish a comparison framework with results obtained from similar surveys on UML usage. The literature review on UML usage led us to formulate the following RQs:

- RQ1: Which is the profile of persons using UML?
- RQ2: In what type of projects is the UML used?
- RQ3: What is the usability and effectiveness of UML diagram types?

Table 1 Studies on UML usage

Authors	Focus of the study	Research types	Research findings
Batra [12]	The need to study UML usage has been identified	Position paper	A set of RQs on UML usage
Dobing and Parsons [13]	A UML usage survey, mainly focusing on the usage of UML diagrams	Survey promoted from OMG with 182 respondents	Findings include which are the most used UML diagrams, the purpose for using UML diagrams, reasons for not using UML diagrams etc
Dobing and Parsons [14]	A UML usage survey focusing on differences on UML diagrams usage on software projects	Web-based questionnaire survey, involving 171 respondents	Differences were found on UML diagrams usage including: frequency, the purposes for which they were used, and the roles of clients/users in their creation and approval. System developers are often ignoring the ‘use case-driven’ approach
Lang and Fitzgerald [15]	To explore what processes, methods and techniques are currently being used for the design of web systems	Literature review and a questionnaire survey, where 164 Irish companies have participated	The results indicated that traditional techniques for web system design are widely employed including UML
Grossman *et al.* [8]	Investigation into the adoption and use of UML	A web-based survey with 150 responses out of 1507 e-mails	Results indicate a wide diversity of opinion regarding the UML, reflecting the relative immaturity of the technology as well as the controversy over its effectiveness
Lange *et al.* [17]	Analysis of common problems with UML models and of techniques for controlling their quality	A user survey and industry case study. Eighty software architects participated, coming from different application domains, and having a different job profile	The UML practices should improve with increased capabilities in development tools for it. Several areas need improvement such as detection of design flaws, omissions and inconsistencies, more uniformity in modelling etc
Wrycza and Marcinkowski [19]	Study on the complexity of UML modelling and investigation of a ‘light’ UML version	A student questionnaire where 180 students participated	They proved that the complexity of teaching and using UML is high and they proposed the introduction of a ‘light’ UML version
Peneva *et al.* [20]	Investigated the utilisation of UML in Bulgarian small and medium companies	A questionnaire that has been administered to more than 100 Bulgarian small and medium enterprises (SME)	The UML is not properly used within the software industry and therefore training is required. The most important topic to be covered by the training is not the UML itself by the methodology to be used
Gu *et al.* [21]	Studies the factors affecting the IT adoption in the context of UML	The research model is based on IT adoption framework and organisational culture theory. Further data were collected from 251 North American organisations	The proposed research model on the adoption of UML is validated

- RQ4: What is the perceived usefulness of UML language?
- RQ5: To what extent are CASE tools used?

The developed questionnaire was composed of 24 questions and it was organised in four sections. The first section included questions that were related with the profile of the individuals who participated in the study. Participants were asked about their education level, their experience and their current job. The second section assessed the projects the participants they have worked on: the size of the project, their role in the project, the type of system that was developed and the application area of the system. Since most participants were involved in more than one project, they were asked to describe 'the most representative UML project', they have worked on. The third section in included questions related with UML usage. It was asked how UML was used in projects where users had to select between 'never, not often, in some cases, frequently and always' for all of the subsequent cases: (a) for model driven development, (b) to explain business needs and business processes, (c) to perform analysis, (d) to perform design, (e) to generate code and (f) to develop test cases.

In the following questions, the participants were asked to describe which of UML diagrams were used in their projects and how they perceived their usefulness. The diagram types that were presented were: activity diagram, class diagram, collaboration or communication diagram, component diagram, deployment diagram, object diagram, package diagram, sequence diagram, state machine diagram, timing diagram, use case diagram and use case narratives. Subsequently, it was asked to define for which task, and at which life cycle phase UML diagram types were used.

The last part of the questionnaire collects various questions related with the use of CASE tools to support software development projects, with the usage of mechanisms such as UML profiles [22], system modelling language (SysML) [23], business process modelling notation (BPMN) [24] etc., the perceived UML usefulness and the benefits that are achieved from UML usage.

The questionnaire was made available online for a period of two months in May and June 2012, via Google Docs© and the URL was distributed to mailing lists of Greek IT professionals (http://www.computer-engineers.gr; http://www.epe.org.gr) and to Hellenic Open University (http://www.eap.gr) students and graduates. The sampling approach used was purposive sampling [25] since we wanted to deliberately select particular subjects for the survey in order to increase the likelihood to produce useful and valid results.

It is estimated that approximately 1000 users received the questionnaire; however, it is not possible to report the exact number of recipients. The questionnaire was answered by 91 individuals, giving a response rate of 9% which is considered as acceptable for this type of exploratory surveys. Although 9% is considered low by the standards of traditionally administered surveys, this is considered acceptable for a web-based industrial survey, where response rates typically cluster about 4–6% [26]. On a similar study, Grossman *et al.* [8] reported a response rate of 10.91%. The issue of 'acceptable' response rate is complex since there is inevitably a high degree of wastage with these surveys (it is impossible to exactly determine the number of e-mails actually arrived, given the spamming algorithms operating in different mail systems) and therefore the calculation of 'response rates' is inconclusive in most cases. The above argument is supported and extensively analysed in the study of Sivo and Saunders [27], who studied the response rates and the validity of inference, in IS questionnaire research.

Two couples of similar and two couples of contradictive questions were incorporated in the questionnaire, in order to examine and secure the reliability of the answers. Those who had more than two mistaken couples of questions were considered that had completed the questionnaire without giving particular attention and therefore those answers should not be taken into account. However, none of the questionnaires contained more than two mistaken couples of questions. The questions used mostly a Likert like scale and they were analysed in a quantitative way (5-points scale) for producing descriptive statistics [28].

To compare the actual usage and the perceived usefulness of UML diagrams, Wilcoxon parametric test was used [29]. The Wilcoxon Signed-Rank test is a non-parametric test used to compare two matched samples and for assessing whether their population mean ranks differ. It is equivalent to the dependent *t*-test, but it does not assume normality in the data and therefore it can be used when this assumption has been violated.

Further, SPSS software package was used for running statistical test for identifying statistical correlations within the data. More specifically, it was used to compare the actual usage and the perceived usefulness of UML diagrams, using Wilcoxon parametric test.

3.1 Research limitations

Although the research addresses its aims and the RQs set, there are limitations that should be stated explicitly. The main limitation of the study is that the produced survey was administered only to IT professionals in Greece. As it has already stated, the majority of Greek companies are considered small and medium enterprises having as main activity, IT service delivery. Therefore the results that will be presented in the next section cannot be safely generalised at the global level. Secondly, the research approach followed in this study is mainly quantitative, which implies that it is very difficult to address the qualitative issues of the software development process, which constitute an important source of tacit knowledge. Finally, the response rate of this study can be considered as a limitation even though there is significant evidence that for internet surveys the obtained response rate is acceptable.

4 Research findings

Although the distributed questionnaire did not have any prerequisite for participation, all participants had UML knowledge. Concerning the educational level of the participants (Table 2), the study indicated that 47% of the population holds a Bachelor degree, 22% a Master degree and an 8% have a Ph.D. Graduate students were considered

Table 2 Educational level

Educational levels	Frequency	Per cent
Bachelor degree	42	47.2
Master degree	20	22.5
Ph.D.	7	7.9
Students	20	22.4
Total	89	100.0

doi: 10.1049/iet-sen.2013.0031

Table 3 Work profile

Work profiles	Frequencies	Per cent
Employee in the private sector	31	34.8
Employee in the public sector	35	39.3
Self-employed	18	20.2
Student	2	2.2
Unemployed	3	3.4
Total	89	100.0

as already having a bachelor degree and therefore they were counted in the 'Bachelor degree' category.

Concerning RQ1, the professional profile of the participants is presented in Table 3 and it illustrates that the population sample is almost evenly distributed in the three main categories of employment: working for the public sector, working for the private sector, self-employed, student or unemployed. Obviously, a number of students are working before graduation.

The experience of the participants in software engineering – in general – was delineated as novice (having <2 years of experience), moderate (having between 2 and 5 years) and expert (having more than 5 years of experience). Table 4 indicates that 49% of the participants could be considered as experts in software engineering. The percentage, in reality, is higher since 20 of the participants in this study are students and 10 out of 20 students stated that they have <2 years of experience. This outcome is important since it indicates that the questioned sample is rather qualified to assess the use of UML in software engineering.

Participants' acquired experience on UML was measured by the number of completed, or almost completed, software projects that they had participated during the past 5 years (see Table 5). According to the findings, each participant has participated approximately in 5 projects and the total number of executed projects for all participants is approximately 413 projects. Among those 412 projects, UML was used in 190 projects, which implies that 46% of the software development projects are using UML. In the similar study of Dobing and Parsons [13], 'respondents reported having been involved in an average of 27 projects (about 6.2 using UML), over an average 15-year career (4.7 using UML) in information technology.' The resulted 46% rate, in this survey, regarding UML utilisation in software projects is significantly bigger that the 23% that was reported in the literature review. This divergence provides an indication that UML usage increased significantly during the recent years, since Dobing and Parsons study is based on data collected from March 2003 to March 2004 and it covers an average 15-year career. This can be explained as well by the facts that (a) the survey of Dobing and Parsons covers a fifteen years period, ending 2004, leads to a much lower usage percentage, since the UML was only introduced in 1997 and (b) results obtained in these two surveys are not directly comparable since they cover a totally different geographical area and generally a significant different population sample. The second part of Table 5 presents the number of projects, where UML was used in relation with the total number of software development projects. From analysing these results, we can conclude that 34% of the participants were solely using UML in their software development projects, since they applied UML in all their projects during the past 5 years.

The primary function of the participants with regard to the most representative UML project they have participated is presented in Table 6. In this question, we have allowed to the participants to select only one answer, since the size of Greek IT companies is medium to small and most of the

Table 4 Participants years of experience in software engineering

Years of experience in software engineering	Frequencies	Per cent	Cumulative per cent
Less than two years	28	31.5	31.5
From two to five years	17	19.1	50.6
More than five years	44	49.4	100.0
Total	89	100.0	

Table 5 Completed software project/project where UML was used

	Number of completed software projects		Number of projects where UML was used	
	Frequencies	Per cent[a]	Frequencies	Per cent
None	9		21	23.6
Less than five projects	40	50.0	51	57.3
From five to ten projects	21	26.2	13	14.6
More than ten	19	23.8	4	4.5
Total	89	100.0		

[a]Percentage values were calculated after excluding the 'none' answer

Table 6 Participants job function in relation with UML projects

Job functions	Frequencies	Per cent
Architect/designer	12	13.5
Business analyst	4	4.5
Client or client representative	4	4.5
Database administrator	4	4.5
Other technical support (e.g. networking and security)	3	3.4
Programmer/developer	24	27.0
Project manager (day-to-day responsibility for this project)	10	11.2
Senior manager (overall responsibility across multiple projects)	4	4.5
Training/education	19	21.3
None of the above	5	5.6
Total	89	100.0

Table 7 Size of the most representative project where UML was used

Size of UML projects	Frequencies	Per cent
Very small (approximately 1 person-year)	39	47.6
Small (approximately 5 person-years)	23	28.0
Average (approximately 10 person-years)	14	17.1
Large (approximately 50 person-years)	6	7.3
Total	82	100.0

Table 8 UML application method

	Always		Frequently		In some cases		Not often		Never		Median
For model driven development	10	(13%)	12	(16%)	28	(37%)	13	(17%)	13	(17%)	in some cases (3)
To explain business needs and business processes	11	(14%)	19	(25%)	27	(36%)	14	(18%)	5	(7%)	in some cases (3)
To perform analysis	10	(13%)	17	(22%)	24	(32%)	19	(25%)	6	(8%)	in some cases (3)
To perform design	12	(16%)	27	(36%)	16	(21%)	11	(14%)	10	(13%)	frequently (4)
To generate code	4	(5%)	12	(16%)	21	(28%)	18	(24%)	21	(28%)	not often (2)
To develop test cases	3	(4%)	11	(14%)	20	(26%)	19	(25%)	23	(30%)	not often (2)

employees usually are engaged in more than one role. The results indicate that only few of the participants have roles directly related to software design and the most frequently found profile, in our sample, was this of the ‘programmer’. A general conclusion that we can derive from Table 6 is that UML is used from all different IT roles, which implies that UML is a significant communication vehicle within the software IT community.

Concerning RQ2, the majority of the projects where UML was used were: customer relationship management systems (23.4%), administrative systems (20.6%), information dissemination systems (17.8%), manufacturing systems (15.0%), data mining systems (13.1%), embedded systems (6.5%) and other (3.7%). At the same time, the majority of the software systems developed were either windows based (43.8%) or web based (42.7%), whereas only a few systems were for embedded software development or for mobile application production. Furthermore, in most of the cases, the development involved a new system (56.2%), the enhancement of an existing system followed (11.2%),

Table 9 Actual usage of UML diagrams

	Always		Most projects		In some projects		Not often		Never		Median	% of positive answers[a]
Activity diagram	17	(20%)	16	(19%)	26	(31%)	9	(11%)	16	(19%)	in some cases (3)	(70%)
Class diagram	29	(35%)	23	(27%)	18	(21%)	6	(7%)	8	(10%)	frequently (4)	(83%)
Collaboration or communication diagram	11	(13%)	15	(18%)	22	(26%)	9	(11%)	27	(32%)	in some cases (3)	(57%)
Component diagram	6	(7%)	10	(12%)	17	(20%)	15	(18%)	36	(43%)	not often (2)	(39%)
Deployment diagram	2	(2%)	10	(12%)	16	(19%)	15	(18%)	41	(49%)	not often (2)	(33%)
Object diagram	8	(10%)	17	(20%)	16	(19%)	5	(6%)	38	(45%)	not often (2)	(49%)
Package diagram	4	(5%)	7	(8%)	12	(14%)	16	(19%)	44	(52%)	never (1)	(27%)
Sequence diagram	14	(17%)	19	(23%)	18	(21%)	14	(17%)	19	(23%)	in some cases (3)	(61%)
State machine diagram	2	(2%)	8	(10%)	14	(17%)	22	(26%)	38	(45%)	not often (2)	(29%)
Timing diagram	3	(4%)	10	(12%)	12	(14%)	11	(13%)	48	(57%)	never (1)	(30%)
Use case diagram	27	(32%)	22	(26%)	14	(17%)	5	(6%)	16	(19%)	frequently (4)	(75%)
Use case narratives	5	(6%)	11	(13%)	20	(24%)	5	(6%)	43	(51%)	never (1)	(43%)

[a]Sums the percentages of positive answers in categories ‘always’, ‘in most projects’ and ‘in some projects’.

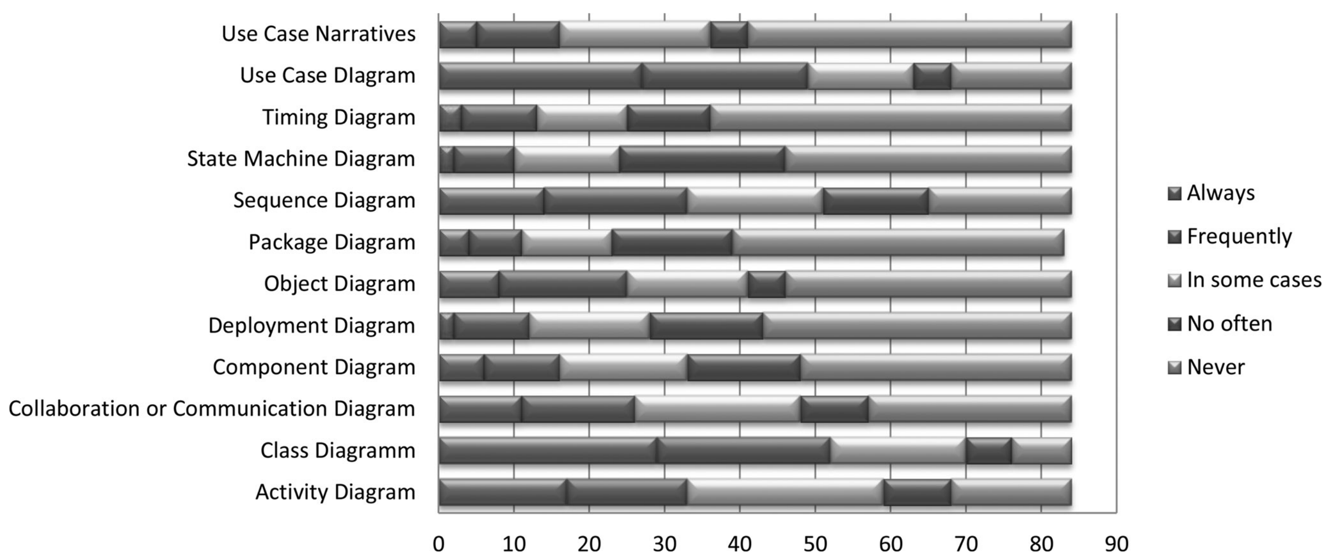

Fig. 1 *Histogram of actual usage of UML diagrams*

IET Softw., pp. 1–11
doi: 10.1049/iet-sen.2013.0031

Table 10 Ranking table for comparing actual usage of UML diagrams and perceived usefulness

		N	Mean rank	Sum of ranks
[PERUSE_activity diagram] – [activity diagram]	Negative ranks	10[a]	18.90	189.00
	Positive ranks	22[b]	15.41	339.00
	Ties	39[c]		
	Total	71		
[PERUSE_class diagram] – [class diagram]	Negative ranks	6[d]	13.00	78.00
positive ranks	17[e]	11.65	198.00	
	Ties	54[f]		
	Total	77		
[PERUSE_collaboration or communication diagram] – [collaboration or communication diagram]	Negative ranks	11[g]	14.95	164.50
	Positive ranks	16[h]	13.34	213.50
	Ties	41[i]		
	Total	68		
[PERUSE_component diagram] – [component diagram]	Negative ranks	9[j]	14.67	132.00
	Positive ranks	17[k]	12.88	219.00
	Ties	37[l]		
	Total	63		
[PERUSE_deployment diagram] – [deployment diagram]	Negative ranks	5[m]	9.50	47.50
	Positive ranks	21[n]	14.45	303.50
	Ties	39[o]		
	Total	65		
[PERUSE_object diagram] – [object diagram]	Negative ranks	10[p]	12.20	122.00
	Positive ranks	14[q]	12.71	178.00
	Ties	40[r]		
	Total	64		
[PERUSE_package diagram] – [package diagram]	Negative ranks	5[s]	11.70	58.50
	Positive ranks	20[t]	13.33	266.50
	Ties	40[u]		
	Total	65		
[PERUSE_sequence diagram] – [sequence diagram]	Negative ranks	12[v]	14.04	168.50
	Positive ranks	16[w]	14.84	237.50
	Ties	41[x]		
	Total	69		
[PERUSE_state machine diagram] – [state machine diagram]	Negative ranks	4[y]	10.00	40.00
	Positive ranks	20[z]	13.00	260.00
	Ties	40[aa]		
	Total	64		
[PERUSE_timing diagram] – [timing diagram]	Negative ranks	3[ab]	14.50	43.50
	Positive ranks	26[ac]	15.06	391.50
	Ties	34[ad]		
	Total	63		
[PERUSE_use case diagram] – [use case diagram]	Negative ranks	6[ae]	11.08	66.50
	Positive ranks	15[af]	10.97	164.50
	Ties	52[ag]		
	Total	73		
[PERUSE_use case narratives] – [use case narratives]	Negative ranks	3[ah]	7.00	21.00
	Positive ranks	15[ai]	10.00	150.00
	Ties	44[aj]		
	Total	62		

[a][PERUSE_activity diagram] < [activity diagram]; [b][PERUSE_activity diagram] > [activity diagram]; [c][PERUSE_activity diagram] = [activity diagram]; [d][PERUSE_class diagram] < [class diagram]; [e][PERUSE_class diagram] > [class diagram]; [f][PERUSE_class diagram] = [class diagram]; [g][PERUSE_collaboration or communication diagram] < [collaboration or communication diagram]; [h][PERUSE_collaboration or communication diagram] > [collaboration or communication diagram]; [i][PERUSE_collaboration or communication diagram] = [collaboration or communication diagram]; [j][PERUSE_component diagram] < [component diagram]; [k][PERUSE_component diagram] > [component diagram]; [l][PERUSE_component diagram] = [component diagram]; [m][PERUSE_deploSTableFootnoteyment diagram] < [deployment diagram]; [n][PERUSE_deployment diagram] > [deployment diagram]; [o][PERUSE_deployment diagram] = [deployment diagram]; [p][PERUSE_object diagram] < [object diagram]; [q][PERUSE_object diagram] > [object diagram]; [r][PERUSE_object diagram] = [object diagram]; [s][PERUSE_package diagram] < [package diagram]; [t][PERUSE_package diagram] > [package diagram]; [u][PERUSE_package diagram] = [package diagram]; [v][PERUSE_sequence diagram] < [sequence diagram]; [w][PERUSE_sequence diagram] > [sequence diagram]; [x][PERUSE_sequence diagram] = [sequence diagram]; [y][PERUSE_state machine diagram] < [state machine diagram]; [z][PERUSE_state machine diagram] > [state machine diagram]; [aa][PERUSE_state machine diagram] = [state machine diagram]; [ab][PERUSE_timing diagram] < [timing diagram]; [ac][PERUSE_timing diagram] > [timing diagram]; [ad][PERUSE_timing diagram] = [timing diagram]; [ae][PERUSE_use case diagram] < [use case diagram]; [af][PERUSE_use case diagram] > [use case diagram]; [ag][PERUSE_use case diagram] = [use case diagram]; [ah][PERUSE_use case narratives] < [use case narratives]; [ai][PERUSE_use case narratives] > [use case narratives]; [aj][PERUSE_use case narratives] = [use case narratives].

whereas complete replacement of an existing system was lower (10.1%) etc. The size of the described – most representative – UML project is presented in Table 7. The percentages presented with regard to UML project size are consistent with the small software projects of the Greek market and with the small size of software companies [9].

The UML application method is presented in Table 8. In this question, we have done an attempt to identify the primary purpose for using UML. Questionnaire participants were presented with choices mainly to select for which phase of software development life cycle the UML was employed. In Table 8, a scale for 1–5 was used, where

Table 11 Wilcoxon signed ranks test statistics

	Z	Asymptotic signed (two-tailed)
[PERUSE_activity diagram] – [activity diagram]	−1.471[a]	0.14
[PERUSE_class diagram] – [class diagram]	−1.918[a]	0.05
[PERUSE_collaboration or communication diagram] – [collaboration or communication diagram]	−0.625[a]	0.53
[PERUSE_component diagram] – [component diagram]	−1.160[a]	0.25
[PERUSE_deployment diagram] – [deployment diagram]	−3.394[a]	0.00
[PERUSE_object diagram] – [object diagram]	−0.851[a]	0.40
[PERUSE_package diagram] – [package diagram]	−2.932[a]	0.00
[PERUSE_sequence diagram] – [sequence diagram]	−0.822[a]	0.41
[PERUSE_state machine diagram] – [state machine diagram]	−3.344[a]	0.00
[PERUSE_timing diagram] – [timing diagram]	−3.920[a]	0.00
[PERUSE_use case diagram] – [use case diagram]	−1.766[a]	0.07
[PERUSE_use case narratives] – [use case narratives]	−2.941[a]	0.00

[a]Based on negative ranks.

number one represents the option 'never' and number five represents the option 'always'. The results show that UML is used mostly for software application design and less for producing code and for specifying test cases. However, it is not possible to derive from Table 8 information, if another language or method was employed, whenever UML was not used, since this information was not included in the questionnaire.

One of the most important questions (RQ3) of this study was which of UML diagram is considered as the most useful in the software development process. According to the findings, the 'class diagram' is the most popular type of diagram, since 83% of the users are using it in some of their projects. The second choice is the 'use case diagram' scoring a 75% and the third is the activity diagram with 70%. The less used diagrams are the 'package diagrams' (27%), the 'timing diagrams' (30%) and the 'state machine diagrams' (29%). The fact that the vast majority of software projects (75%) that are considered in this survey, are characterised as 'very small' and 'small', obviously affects the usage pattern for these three diagram types, since these diagrams are mostly used in large and complex software development projects.

Similar findings are reported in Dobing and Parsons study [14], where 'class diagram' is the most frequently used component with 73% of respondents saying they were used in two-thirds or more of their projects, followed again by 'use case diagrams'. In Lang and Fitzgerald [15] study, that studied web development projects, reports usage for 'use case diagrams/scenarios' 72% and for 'class diagrams' 62%. The findings are presented in detail in Table 9 and in Fig. 1.

RQ4 concerning the perceived usefulness of UML language in relation with its actual use is presented in Table 10. In Table 10, the variable with prefix PERUSE represents the PERceived USEfulness of each different UML diagram that is compared with its actual use. From Table 10, in the first row that refers to the activity diagrams, we can see that ten participants have lower perceived usefulness score in comparison with the score for actual use, whereas 22 participants consider that the perceived usefulness of activity diagrams is higher than the actual use. Similarly, we can interpret the data presented in the subsequent rows.

To test the hypothesis that

'H_o: UML diagram usage = perceived usefulness of UML diagram'

'H_1: UML diagram usage ≠ perceived usefulness of UML diagram'

We need to test which of the results represent a statistical significant difference. A confidence level of 95% was used which implies an alpha value of $a = 0.05$ although in practice alpha values of 0.1 (10%) or 0.01 (1%) are commonly used as well.

For our study, we have selected that the rejection region of the null hypothesis is for 'p-values' ≤ 0.05. In Table 11, we can see that the p value is ≤ 0.05 for class diagrams, deployment diagrams, package diagrams, state machine diagrams, timing diagrams and use case narratives. For these cases, we can reject the null hypothesis, implying that UML 'diagram usage is different than perceived usefulness of UML diagram. For these diagrams that we have statistically significant results, the perceived usefulness is higher than the actual usage and therefore they should be used more (since positive ranks no substantial higher than negative).

In Table 12, we present which UML diagram is used in each stage of a typical software life cycle. The main conclusion that can be derived from the obtained results is that UML is mostly used in the early phases of software life

Table 12 Usage of UML diagrams in software development

	Business analysis		Analysis		Design		Coding		Testing	
Activity diagram	11	(18%)	20	(32%)	23	(37%)	6	(10%)	2	(3%)
Class diagram	3	(4%)	10	(15%)	35	(52%)	19	(28%)	0	(0%)
Collaboration or communication diagram	10	(20%)	12	(24%)	16	(32%)	9	(18%)	3	(6%)
Component diagram	4	(8%)	15	(31%)	22	(45%)	7	(14%)	1	(2%)
Deployment diagram	5	(11%)	8	(18%)	16	(36%)	12	(27%)	4	(9%)
Object diagram	4	(8%)	10	(20%)	19	(39%)	14	(29%)	2	(4%)
Package diagram	6	(12%)	10	(20%)	17	(35%)	11	(22%)	5	(10%)
Sequence diagram	7	(12%)	11	(19%)	25	(44%)	10	(18%)	4	(7%)
State machine diagram	4	(8%)	13	(25%)	22	(43%)	7	(14%)	5	(10%)
Timing diagram	9	(20%)	11	(24%)	13	(28%)	5	(11%)	8	(17%)
Use case diagram	17	(25%)	20	(30%)	21	(31%)	7	(10%)	2	(3%)
Use case narratives	12	(26%)	10	(21%)	18	(38%)	6	(13%)	1	(2%)

doi: 10.1049/iet-sen.2013.0031

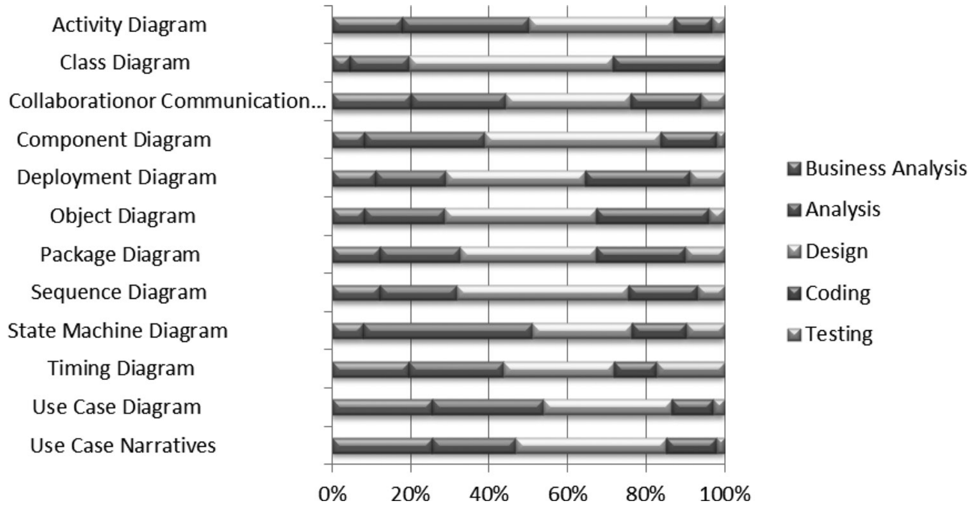

Fig. 2 *Usage histogram for UML diagrams per phase*

cycle and mostly in analysis and design phases. More specifically for the business analysis phase, the most popular diagram is the 'use case diagram', for the analysis phase the most frequently used diagrams are the 'use case diagram' and the 'activity diagram'. This implies that users prefer 'activity diagrams' for documenting their use cases, rather than use case narratives. The 'class diagram' is considered more useful for the design and coding phases, whereas in testing the more applicable diagrams are found to be the 'timing diagram', the 'state machine diagram' and the 'package diagram. The same results are represented graphically in Fig. 2.

Concerning RQ5 and the adoption of CASE tools to support the use of UML, 71% of participants are using CASE tools, mostly for design – in 88% of the cases – and for analysis – in 84% of the cases. On the contrary, CASE tools for testing are used in 66% of the cases.

During the evolution of UML, a number of extension mechanisms were developed, and the language was customised for specific application domains. As such, the participants were asked (see Table 13) if they use these mechanisms (among the most popular available). A large percentage of the user group is not familiar at all with these technologies (e.g. SysML and XML metadata interchange (XMI)), whereas others rarely or never use (negative answers) them. Concluding the most popular mechanism is the BPMN.

In the question 'How do you rate the success of these typical recent UML projects you have participated?' users replied that 60% of the projects were successful or very successful and only 13% of the projects were unsuccessful. The rest of the answers were impartial.

Concerning the overall opinion of the users for UML, most users expressed positive opinions regarding faster system

Table 13 Usage of other UML features

	Positive answers		Negative answers		Not familiar	
SysML	7	(12%)	21	(36%)	30	(52%)
BPMN	15	(26%)	22	(38%)	21	(36%)
domain specific modelling languages	17	(29%)	20	(34%)	21	(36%)
XMI (for model storage or interchange)	18	(31%)	18	(31%)	22	(38%)
UML profiling mechanism	22	(36%)	20	(33%)	19	(31%)

Table 14 Overall opinion on UML usage

	Agree		Somehow agree		Neutral		Somehow disagree		Disagree		Median
Results in higher quality systems	26	(29%)	34	(38%)	18	(20%)	8	(9%)	3	(3%)	Somehow agree (4)
Improves our productivity when building systems	31	(35%)	29	(33%)	19	(21%)	8	(9%)	2	(2%)	Somehow agree (4)
Enables us to build systems more quickly	25	(28%)	27	(30%)	24	(27%)	10	(11%)	3	(3%)	Somehow agree (4)
Helps ensure that the final system meets the functional requirements	38	(43%)	27	(30%)	17	(19%)	4	(4%)	3	(3%)	Somehow agree (4)
Helps in ensuring that the clients and users understand what system analysts are proposing	23	(26%)	32	(36%)	20	(22%)	12	(13%)	2	(2%)	Somehow agree (4)
Helps to estimate the size of the system to be developed	28	(31%)	31	(35%)	21	(24%)	7	(8%)	2	(2%)	Somehow agree (4)
Helps to decrease the cost of enhancements to the system	22	(25%)	23	(26%)	24	(27%)	15	(17%)	5	(6%)	Somehow agree (4)
Improves the communication within project	31	(35%)	36	(40%)	14	(16%)	6	(7%)	2	(2%)	Somehow agree (4)
Assists developers in performing their job	35	(39%)	29	(33%)	17	(19%)	6	(7%)	2	(2%)	Somehow agree (4)
Is complex and difficult to use	6	(7%)	16	(18%)	23	(26%)	25	(28%)	19	(21%)	Neutral (3)

doi: 10.1049/iet-sen.2013.0031

building (75%), for higher quality systems (73%) and for system enhancements' cost-decrease (72%). To the question 'if they consider the UML complex and difficult to use', 49% expressed a negative position (disagree or somehow disagree), 25% expressed a positive position (that they agree with the statement) and 26% an impartial neutral position. In Table 14, the detailed results are illustrated. A scale for 1–5 was used, where number one represents the option 'disagree', number two the option 'somehow disagree', number three the option 'neutral', number four the option 'somehow agree' and finally number five represents the option 'agree'.

5 Conclusions

The results of the study demonstrate that UML, more than 15 years after its introduction, is still widely used. Even though the UML is so widely used, only a few studies can be found in the literature documenting how UML is used and under which conditions.

In this study, we have attempted to contribute towards this direction and to document in what type of projects is used, which software development methodology is used, where each UML diagram is used, if it is perceived useful etc.

The survey was administered to IT professionals in Greece, it covered all areas of software development and all different types of systems (web based, windows based etc.). The findings indicate that, despite its limitations, UML is used in the majority of software development projects, out of the total amount of 413 projects that were reported in this study, the UML was used in almost half of these projects. Among the UML diagrams, the most popular is considered to be the 'class diagram', followed in popularity by the 'use case diagram'. The least used diagrams are 'package diagrams', 'timing diagrams' and 'state machine diagrams'. Findings indicate that UML is mostly used for analysis and design rather for coding and testing. Even though the UML is extensively used, UML extensions such as SysML, XMI are not well known and a large percentage of the user group is not familiar, whereas others rarely or never use (negative answers) them. In relation to the benefits from using UML, users replied that UML enables them to build systems more quickly; that it results in higher quality systems; that it helps to decrease the cost of enhancements to the system and they consider their UML projects successful or very successful. Finally, users declare that they will use again the UML in their future projects and they expect that UML usage will increase further during the next years.

The main conclusion is that despite the limitations and extensions needed, the UML is the only general-purpose modelling language that is an industry standard for specifying software-intensive systems, that is supported by numerous tool vendors, it is integrated within integrated development environments and at the same time is widely known and supported and the same time software development community recognises its value.

As next steps of this study, we indent to investigate how UML is used in relation with different project types and different job profile in an attempt to understand how UML is used in various domains and by various professionals.

6 Acknowledgment

The research presented in this paper has been co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Programme 'Education and Lifelong Learning' of the National Strategic Reference Framework. In particular, the research work has been co-financed by the R&D programs 'ONSOCIAL' and 'SPRINT SMEs' which take place in the context of the 'ARCHIMEDES III' National Research Programme. The authors thankfully express their appreciation of the valuable support to all respondents and especially to those of them who propagated the questionnaire.

7 References

1 Booch, G.: 'Object-oriented analysis and design with applications' (Benjamin/Cummings, 1994)
2 Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: 'Object-oriented software engineering: a use case driven approach' (Addison-Wesley, Reading, MA, 1992)
3 Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.: 'Object-oriented modeling and design' (Prentice-Hall, 1991)
4 OMG, 'Unified Modeling Language: Superstructure Version 2.4.1, Object Management Group', Available at http://www.omg.org/spec/UML/2.4.1/, accessed October 2012
5 Booch, G., Rumbaugh, J., Jacobson, I.: 'The UML reference manual' (Addison-Wesley, Boston, MA, 2004, 2nd edn.)
6 Agarwal, R., Sinha, A.P.: 'Object-oriented modeling with UML: a study of developers' perceptions', *Commun. ACM*, 2003, **46**, (9), pp. 248–256
7 Krogstie, J.: 'Evaluating UML using a generic quality framework', in Favre, L. (Eds.): 'UML and the unified process' (IRM Press, London, 2003)
8 Grossman, M., Aronson, J.E., McCarthy, R.V.: 'Does UML make the grade? Insights from the software development community', *Inf. Soft. Technol.*, 2005, **47**, pp. 383–397
9 Kostoglou, V., Paparrizos, K., Zafiropoulos, K.: 'Investigating human resource management policies of the ICT labour market', *Oper. Res. Int. J.*, 2004, **4**, (1), pp. 57–72
10 Budgen, D., Burn, A.J., Brereton, O.P., Kitchenham, B.A., Pretorius, R.: 'Empirical evidence about the UML: a systematic literature review', *Softw. – Pract. Exp.*, 2011, **41**, pp. 363–392
11 Batra, D.: 'Unified modeling language (UML) topics: the past, the problems, and the prospects', *J. Database Manage.*, 2008, **19**, (1), pp. i–vii
12 Batra, D.: 'Unified modeling language (UML) topics: cognitive issues in UML research', *J. Database Manage.*, 2009, **20**, (1), pp. i–x
13 Dobing, B., Parsons, J.: 'How UML is used', *Commun. ACM*, 2006, **49**, (5), pp. 109–113
14 Dobing, B., Parsons, J.: 'Dimensions of UML diagram use: a survey of practitioners', *J. Database Manage.*, 2008, **19**, (1), pp. 1–18
15 Lang, M., Fitzgerald, B.: 'New branches, old roots: a study of methods and techniques in web/hypermedia systems design', *Inf. Syst. Manage.*, 2006, **23**, (3), pp. 62–74
16 Goodhue, D.L.: 'Development and measurement validity of a task technology fit instrument for user evaluations of information systems', *Decis. Sci.*, 2008, **29**, (1), pp. 105–138
17 Lange, C., Chaudron, M., Muskens, J.: 'In practice: UML software architecture and design description', *IEEE Softw.*, 2006, **23**, (2), pp. 40–46
18 Kruchten, P.: 'The 4 + 1 view model of architecture', *IEEE Softw.*, 1995, **12**, (6), pp. 45–50
19 Wrycza, S., Marcinkowski, B.: 'Towards a light version of UML 2.X: appraisal and model', *Organizacija*, 2007, **40**, (4), pp. 171–179
20 Peneva, J., Ivanov, S., Tuparov, G.: 'Utilization of UML in Bulgarian SME – possible training strategies'. Int. Conf. on Computer Systems and Technologies – CompSysTech, 2006
21 Gu, V.C., Cao, Q., Duan, W.: 'Unified modeling language (UML) IT adoption — a holistic model of organizational capabilities perspective', *Decis. Support Syst.*, 2012, **54**, (1), pp. 257–269
22 Fuentes, L., Vallecillo, A.: 'An introduction to UML profiles', *Eur. J. Inf. Prof.*, 2004, **5**, (2), pp. 6–13
23 Friedenthal, S., Moore, A., Steiner, R.: 'A practical guide to SysML: the systems modeling language' (Morgan Kaufmann/Elsevier, Waltham, MA, USA, 2011, 2nd edn.)
24 OMG: 'Business Process Model and Notation (BPMN), version 2', Object Management Group, Available at http://www.omg.org/spec/BPMN/2.0, accessed November 2012.

doi: 10.1049/iet-sen.2013.0031

25 Denscombe, M.: 'The good research guide for small-scale social research projects' (McGraw-Hill, Open University Press, NY, 2007, 3rd edn.)
26 Rowlands I., Olivieri R.: 'Overcoming the barriers to research productivity', Publishing research Consortium, Available at http://www.publishingresearch.net/documents/overcoming_barriers_report.pdf, accessed May 2013
27 Sivo, S., Saunders, C.: 'How low should you go? Low response rates and the validity of inference in IS questionnaire research', *J. Assoc. Inf. Syst.*, 2006, **7**, (6), pp. 351–414
28 Saunders, M., Lewis, P., Thornhill, A.: 'Research methods for business students' (Financial Times Prentice-Hall, England, 2009, 5th edn.)
29 Field, A.: 'Discovering statistics using SPSS' (Sage Publications, England, 2006, 2nd edn.)

doi: 10.1049/iet-sen.2013.0031