State of Component Models Usage: Justifying the Need for a Component Model Selection Framework

Hazleen Aris[1] and Siti Salim[2]
[1]College of Information Technology, Universiti Tenaga Nasional, Malaysia
[2]Faculty of Computer Science and Information Technology, University of Malaya, Malaysia

Abstract: *The presence of a large number of component models has caused some difficulties in selecting suitable component models to be used, either for research purpose or for software development purpose. Lack of framework or standard that can be used to guide the process of selecting suitable component models is believed to be one of the reasons that have caused the difficulties. Therefore, in this article, the need for a component model selection framework is justified. The selection framework can be applied by the software developers to help them determine suitable component models to be used in their software development projects. Possible contributions of the framework to the research and software development industry communities are also identified. These are achieved by examining the current state of component models usage in both research and software development industry communities, which are obtained from a number of related resources found from exhaustive literature search.*

Keywords: *Component model usage, component model selection, selection framework, component model survey, component model review, and component model state.*

Received July 3, 2009; accepted November 5, 2009

1. Introduction

Component-Oriented Software Development (COSD) is an approach towards software development where software applications are produced by composing software components. A software component is defined as "a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard" [7]. From the definition, we can clearly see the need for software components to conform to a component model in order to allow them to be independently deployed and composed as is i.e. achieve the purpose of their creation.

According to Lüer and van der Hoek [15], "a component model is a combination of (a) a component standard that governs how to construct individual components and (b) a composition standard that governs how to organise a set of components into an application and how those components globally communicate and interact with each other." Conformance to a component model is one of the properties that distinguish components from other forms of packaged software [4]. Therefore, component model plays an important role in COSD.

Due to its importance, component model is listed as one of the challenges to be addressed in Component-Based Software Engineering (CBSE) [9], and classified as one of the CBSE research areas [12]. Lau and Wang [13] also regarded component model as the cornerstone of any COSD methodology and to Aris and Salim [3], it is the essence that largely determines the success of COSD.

Therefore, continuous effort is being spent on the development of component models. Each effort is attempting to fulfil the expectations of a component model, hence fulfilling the ultimate aim of COSD – to develop applications by composing plug-and-play components.

Research intensity in the development of component models has resulted in the origin of a plethora of component models [8]. The presence of a large number of component models has caused some difficulties to the component developers to choose suitable component models to be used in their software development projects [8]. As a result, despite the availability of many component models, only a few common ones are being frequently used while others are hardly applied once developed.

We believe that the presence of a framework that can guide the software developers in determining the most suitable component models to be used will be able to rectify the situation. Such a framework should allow software developer to specify desired characteristics of the application to be developed and based on those characteristics, the framework will be able to recommend suitable component model(s) from a pool of available component models.

Therefore, the objectives of this research are (1) to justify the need for such a framework that we call a Component Model Selection (CMS) framework, and (2) to identify the contributions of the framework to the research and software development industry communities. In the course of fulfilling the objectives, this research attempts to satisfy the following four research questions, labelled Q_1 to Q_4.

Q_1. What are the available component models to date?
Q_2. Which of the component models are being applied by the component developers?
Q_3. How do the component models being selected for application?
Q_4. Is there any framework/standard followed in making the selection?

In this article, answers to the above questions are presented and discussed.

2. Method

For each of the research questions listed in the previous section, the scope of investigation is divided into two; the research community and the industrial community. Research community applies components and component models for research purposes while industrial community applies them for business purposes.

To obtain the required information from the research community, a literature search was performed to identify research work that contains information on the existing, available component models. Four such literatures were found and detail review was then performed on these literatures. Summaries of each literature are presented in section 3 of this article.

Similar method was also used to obtain information from the industrial community. From the literature search, two survey results that are relevant to the context of our research were found. Both survey results contain the needed information on the application of component-oriented software development in the software development industry.

The first survey with the title 'CBSE State of Practice and Experience Survey' [10] was an online survey conducted to identify the state of CBSE practice and hence, determining its future challenges. The second survey was carried out to investigate the current state of component-oriented software development [2], which aimed at identifying the potential of COSD application, the problems faced in applying COSD and the factors affecting its application. Discussions on the relevant results from both surveys are elaborated in section 4 of this article.

3. Component Models in Research

The presence of a large number of available component models has enabled researchers to perform reviews on the existing component models. The reviews are usually in the form of comparative analyses that study the state of the component models from a specific point of view. The following four subsections provide summaries on four reviews found. The summaries are consistently structured to provide information in the following order.

- Problem statement that motivates the review
- Objective of the review
- Total of component models reviewed
- Basis for including the selected component models in the review

3.1. Review 1 – Component Models for Embedded Systems [1]

In this review, the authors argued that common component technologies such as EJB, Java Beans, COM and .Net, which are mainly used for desktop and distributed enterprise applications, are not being applied to the development of embedded systems. Embedded systems are often the combination of:

- resource constrained systems,
- safety critical systems and
- real-time systems.

Common component technologies are not suitable for resource constrained systems because they are simply too demanding both in computing power and memory. They are not suitable for safety critical systems because it is hard to verify their functionalities due to their complexity and black box property. They are also not suitable for real-time systems because they rely on unpredictable dynamic binding and are optimised for average case performance rather than worst case performance.

Therefore, the review aimed at providing descriptions of a collection of component models, from both academia and industry, which are deemed suitable to be applied in the development of embedded systems. A total of 8 component models were included in the descriptions. The component models are CCM, IEC 61131, Koala, PBO, PECOS, PECT/Pin, Robocop and Rubus.

The component models were selected based on the availability of sufficient written information with the respective author's claim that the technology is suitable for embedded systems and the need to achieve a combination of research-based and industry-based component technologies.

3.2. Review 2 – Taxonomy for Software Component Models [13]

In this review, a taxonomy of component models based on component composition was proposed. Taxonomies based on component semantics and syntax were also considered in the review. Nevertheless, taxonomy based on component composition was discovered to be the most practical and relevant. Using the taxonomy based on component composition, the study aimed at presenting the state of current component models, which is regarded as not (yet) ideal for fulfilling COSD promise.

A total of 13 component models were included in the review. The component models were selected on the basis that they have reached a mature stage of development with sufficient documentation available. The selected component models are .NET, Acme-like ADL, CCM, COM, EJB, Fractal, JavaBeans, Koala, KobrA, PECOS, SOFA, UML 2.0/UML and Web Services.

The review concluded that all of the above component models attempted to fulfil the characteristics of an idealised component life cycle, but with varying degrees of success. It therefore opens up rooms for improvement of existing component models.

3.3. Review 3 – Classification Framework for Component Models [8]

The diversity of component models with differing characteristics, from both industry and research communities, has made it more difficult to properly understand the component-based principles, to properly select a component model of interest and to compare component models [8].

Therefore, in this review, a multi-dimensional classification framework was proposed with the aim of identifying and quantifying the basic principles of component models. The dimensions under consideration are lifecycle, constructs, extra-functional properties and domains [8]. Each dimension was further refined into a number of characteristic points to enable detail study of each dimension.

The classification was performed by identifying which component models support which characteristic points under the four dimensions of the framework. If they support a particular characteristic point, information on how the support is achieved was also included.

A total of 14 component models were classified using the framework. The component models are Autosar, BIP, CCM, COM, EJB, Fractal, Koala, OSGi, PECOS, PECT/Pin, Robocop, Rubus, SaveCCM and SOFA.

These component models were selected because they were regarded as to have fulfilled the minimum criteria of a component model, which means that:

- They explicitly or implicitly identify components and
- They define rules for specification of component properties and means of their composition.

From the classification exercise, the following recurring patterns were observed.

- General component models utilise client-server style,
- Specialised component models mainly use pipe and filter style and
- Support for extra-functional properties is rather scarce.

3.4. Review 4 – Common Component Modelling Example [17]

The final review was conducted in a form of a competition, where a contest was held with the aim of evaluating and comparing the practical application of the component models and their corresponding specification techniques. Competition was chosen as the form of review due to the difficulty in performing comparative study on the existing component models as they differ in terms of formal component modelling and quality prediction employed.

In the competition, textual description and Java implementation of a Common Component Modelling Example (CoCoME), a Trading System [11], were provided and each participating team was required to model the example using their respective component modelling approach.

Each team was given approximately two months to deliver beta version of their modelling of the CoCoME. After that, their modelling was evaluated by other participating teams before the final modelling of the CoCoME was delivered.

From an initial of 19 registered teams, only 13 teams successfully obtained their results in the required form using the following component models respectively; CoIn, Cowch, DisCComp, Focus/AutoFocus, Fractal, GCM/ProActive, Java/A, KLAPER, KobrA, Palladio, SOFA, rCOS and Rich Services.

After delivering their final modelling of the CoCoME, each participating team presented their results in front of an international jury consisting of prominent researchers in the area.

Even though the detailed textual description of the CoCoME and its Java implementation were provided, it was neither possible for the jury to provide any useful comparative analysis of the presented component models, nor was they able to rank the component models according to any order that is useful or informative. In the end, the jury concluded that none of the contesting approaches enables comprehensive all-of-system modelling [5]. Nevertheless, the gap between practical applicability and demonstrated modelling capability exposed by the contest remains significant and leaves much room for further work.

Table 1 below summarises the four reviews described above, retrieving information about:

1. Their objectives,

2. the component models included in the review,
3. criteria used to include the component models and
4. whether any framework/standard is used to select the component models to be reviewed.

Item 2 above provides answer to research questions Q_1 and Q_2. Item 3 answers Q_3 and item 4 answers Q_4.

Table 1. Review summary.

Review Work	Review Objective	Component Models Reviewed	Selection Criteria	Framework/Standard Used for the Selection
Review 1	To provide descriptions of a collection of component models which are deemed suitable to be used in the development of embedded systems	CCM IEC 61131 Koala PBO PECOS PECT/Pin Robocop Rubus	Component models that are suitable for embedded systems development	None
Review 2	To show the state of current component models using taxonomy based on component composition	.NET Acme-like ADL CCM COM EJB Fractal JavaBeans Koala KobrA PECOS SOFA UML 2.0/UML Web Services	Component models that have reached level of maturity with sufficient documentation available	None
Review 3	To develop classification framework that identifies and quantifies the basic principles of component models	Autosar BIP CCM COM EJB Fractal Koala OSGi PECOS PECT/Pin Robocop Rubus SaveCCM SOFA	Component models that • explicitly or implicitly identify components and • define rules for specification of component properties and means of their composition	None
Review 4	To evaluate and compare the practical application of the component models and their corresponding specification techniques	CoIn Cowch DisCComp Focus/AutoFocus Fractal GCM/ProActive Java/A KLAPER KobrA Palladio SOFA rCOS Rich Services	Component models that can specify the given CoCoME in the required form	None

4. Component Models in Industry

Literature search performed to obtain answers that satisfy the research questions from the industrial community scope ended up with two survey results that contain information on the state of component models usage in the industry. The first survey was carried out to identify the CBSE state-of-practice and the second survey was performed to investigate the application of COSD among software developers. These surveys are elaborated in subsections 4.1 and 4.2 respectively.

Other survey results that study the current state of components usage in software development projects were also found [6, 14, 16]. However, information presented on these surveys does not contain the information on the state of component models usage in the industry to answer the research questions.

4.1. Survey 1 – State of CBSE Practice and Experience [10]

This survey aimed to provide the understanding of the landscape of CBSE and hence, to determine the future challenges that will be faced by CBSE. In the survey, an online questionnaire was developed and organisations from all over the world were invited to participate in answering the online questionnaire. As can be seen in Figure 1, organisations from all over the world participated in the survey with the most number of participations came from the European countries.

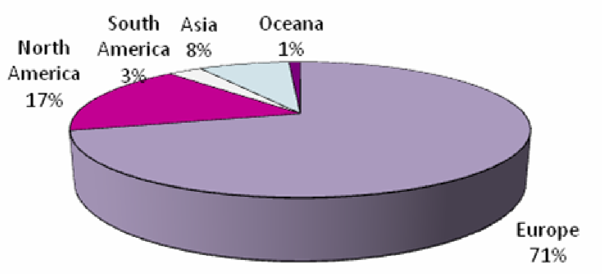

Figure 1. Countries of the participating organisations [10].

Various category of information was captured by the survey and among the information gathered was the type of component models used by the organisations. This information is shown in Figure 2.

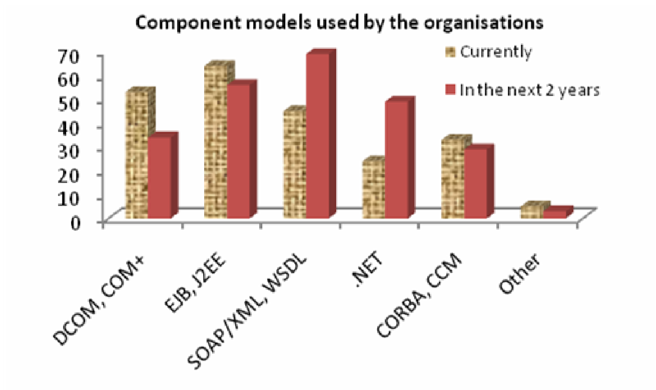

Figure 2. Component models used by the participating organisations [10].

Other than the types of component models used, the survey also gathered the demographic information of the participating organisations, goals of using components, types of project that used components and inhibitors to the use of components.

4.2. Survey 2 – State of COSD Application in Malaysia [2]

The second survey was actually performed as part of our research that studies the current state of COSD application among the software developers in Malaysia [2]. Similar to the first survey, this survey enclosed numerous questions that probed the current state of COSD application. Nonetheless, this article only includes the result of a question that is relevant to its context of discussion, which is on the types of component models used by the software developers as shown in Figure 3 below.

Information on the types of component model used by the industry obtained from both survey results answer research questions Q_1 and Q_2 from the industrial community scope. However, information on the selection criteria and the use of framework in making the selection cannot be found from the survey results to answer Q_3 and Q_4.

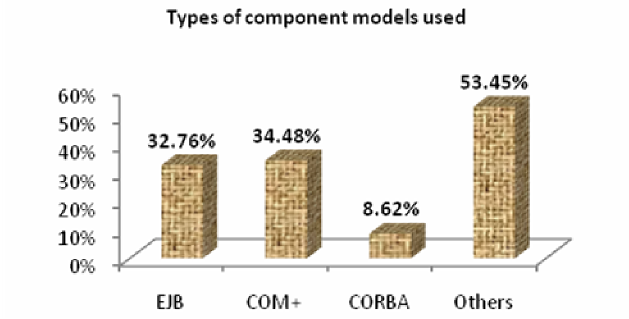

Figure 3. Types of component models used by the software developers.

5. Findings and Discussion

In this section, findings from the reviews from the research community and survey results from the industrial community presented in the previous two sections are analysed. Based on the analysis, the following will be determined.

1. The need to have a Component Model Selection (CMS) framework to assist component developers in selecting suitable component models to be used and
2. The contributions of the CMS framework to the communities (research and industrial).

5.1. Component Models Usage in the Research

Table 2 summarises the component models included in the four reviews described in section 3. Even though Table 2 does not include every single component model that exists, we believe that majority of the existing component models have been included and they sufficiently represent the rest of the component models.

Table 2. Component models and reviews that include them.

Component Model	Review 1	Review 2	Review 3	Review 4
1. .NET		✓		
2. Acme-like ADL		✓		
3. Autosar			✓	
4. BIP			✓	
5. CCM	✓	✓	✓	
6. CoIn				✓
7. COM		✓	✓	
8. Cowch				✓
9. DisCComp				✓
10. EJB		✓	✓	
11. Focus/AutoFocus				✓
12. Fractal		✓	✓	✓
13. GCM/ProActive				✓
14. IEC 61131	✓			
15. Java/A				✓
16. JavaBeans		✓		
17. KLAPER				✓
18. Koala	✓	✓	✓	

Component Model	Review 1	Review 2	Review 3	Review 4
19. KobrA		✓		✓
20. OSGi			✓	
21. Palladio				✓
22. PBO	✓			
23. PECOS	✓	✓	✓	
24. PECT/Pin	✓		✓	
25. rCOS				✓
26. Rich Services				✓
27. Robocop	✓		✓	
28. Rubus	✓		✓	
29. SaveCCM			✓	
30. SOFA		✓	✓	✓
31. UML 2.0/UML		✓		
32. Web Services		✓		

In Figure 4, we plot a graph of the frequency of reviews versus component models to illustrate how frequent each component model is being included in the reviews. From the figure, we can see that none of the component models are included in all of the four reviews. CCM, Fractal, Koala, PECOS and SOFA are the component models that are most frequently reviewed, followed by COM, EJB, KobrA, PECT/Pin, Robocop and Rubus component models. From the corresponding bar chart in

Figure 5, we can also see that the majority of the component models (21 out of 32) are cited by only one review.

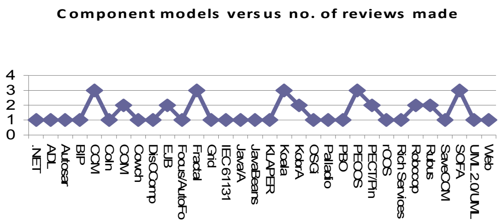

Figure 4. Frequency of the component models reviewed.

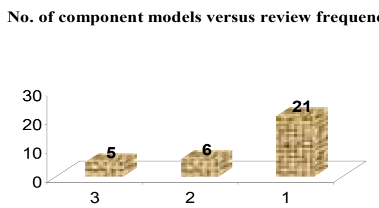

Figure 5. Component models grouped according to number of times they are being reviewed.

With regard to the basis used or reason for including the selected component models for review, it can be observed that the reasons vary from one review to another as seen from Table 1. None of the review followed any framework/standard in making the selection. Therefore, with regard to the use of component models in the research community, the following inferences can be made.

- The commonly used component models are restricted to only a few component models, with majority of the component models proposed by researchers are not acknowledged by others.
- The bases for short-listing component models vary
- No systematic component selection process that properly considers certain aspects or properties of component models is used.

Therefore, if a framework exists, which can be used by the software developers to allow them to (1) thoroughly consider existing component models and (2) guide them in selecting the component models is available, the selection process can be made more systematic and more objective. With this, we justify that there is a need to have a Component Model Selection (CMS) framework for the research community that will contribute to the following areas:

1. able to increase the chances of existing component models being applied by the researchers and
2. enable more systematic selection of component models for research use.

These contributions are illustrated in Figure 6 below.

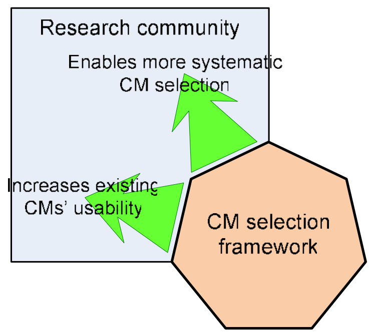

Figure 6. Contributions of the CMS framework to the research community.

5.2. Component Models Usage in the Industry

Based on the findings from the component models usage in the industrial community presented in both survey results, a scenario similar to the research community in terms of the types of component models used can be observed. Majority of the component models applied by the software developers are limited to a small number of component models, which are already well-known in the industry, such as EJB, COM+ and CORBA component models. In fact, the limitation on the choice of component models used is more obvious as seen from Figure 2 and Figure 3. From Figure 2, it shows that only 5 component models are commonly used by the organisations and in Figure 3, only 3 component models are common to the software developers. This indicates that despite the large number of component models available, only a small number component models is applied in the industry.

However, since information on the process of selecting a particular component model for application cannot be obtained from both surveys, no inference can be made with regard to the criteria used in selecting the component models and whether or not framework is used in making the selection.

The presence of CMS framework will enable component developers to consider the widely available component models before making the selection, hence increasing their chances of being used by the component developers. Therefore, to the industrial community, the CMS framework can contribute in the following areas.

1. Able to expose existing component models to the software developers and
2. Enable more systematic selection of component models for industrial use.

These contributions are illustrated in Figure 7 below.

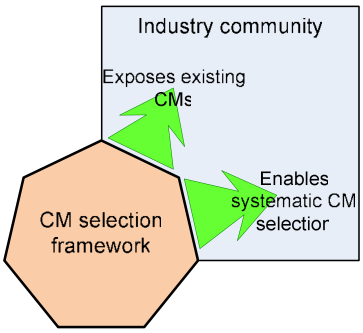

Figure 7. Contributions of the CMS framework to the industry.

A CMS framework is therefore needed for both research and industrial communities. The areas of contribution in both communities are summarised as follows:

1. To increase the opportunities of any existing component models to be used by the software developers,
2. To expose other existing (but less popular) component models and
3. To enable more systematic selection of component models to be used.

These contributions are combined in Figure 8 to illustrate the contributions of the CMS framework in both communities. The third contribution is common to both research and industrial communities. Therefore, it is placed in the middle of both quadrants in Figure 8.

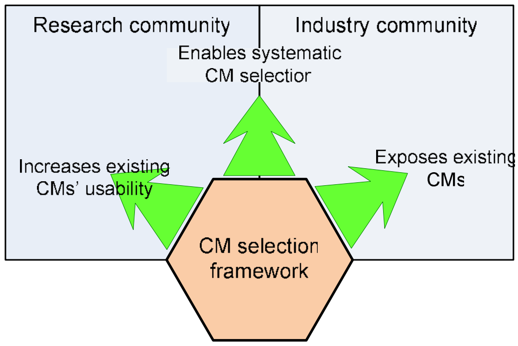

Figure 8. Contributions of the CMS framework to the research and industrial communities.

6. Conclusions

This paper presented and discussed about the current state of component models usage in both the research community and industrial community with the aim of justifying the need and identifying the contributions of a CMS framework to both communities. Investigation from the research community was achieved by thoroughly reviewing research work that performed comparative analyses on the existing component models while investigation from the industrial community was accomplished through the related survey results found. It is discovered that the CMS framework is needed and its existence will enable more systematic selection of suitable component models, expose other existing component models and increase the chances of existing component models to be used by the software developers.

Acknowledgements

We would like to thank the University of Malaya and Universiti Tenaga Nasional for the funds provided for this research under their short term research fund schemes. We would like to also express our appreciation and thanks to the software developers who have participated in the survey. This project cannot be materialised without their cooperation.

References

[1] Åkerholm M. and Fredriksson J., "A Sample of Component Technologies for Embedded Systems," *Technical Report*, Mälardalen Research and Technology Centre, Mälardalen University, Västerås, Sweden, 2004.

[2] Aris H. and Salim S., "Current State of Component-Oriented Software Development Practice in Malaysia: Towards Identifying its Potential and Research Areas," *Journal of Computer Science*, vol. 5, no. 1, pp. , 2009.

[3] Aris H. and Salim S., "Refinement of Component Model Standards and Conventions," *in Proceedings of the 10th International Conference*

on Software Reuse, LNCS 5030, Springer-Verlag Berlin Heidelberg, pp. 278-281, 2008.

[4] Bachmann F., Bass L., Buhman C., Comella-Dorda S., Long F., Robert J., Seacord R., and Wallnau K., "Technical Concepts of Component-Based Software Engineering," *Technical Report*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2000.

[5] Broy M., Siedersleben J., and Szyperski C., "CoCoME Jury Evaluation and Conclusion," *The Common Component Modeling Example Comparing Software Component Models*, LNCS 5153, Springer-Verlag Berlin Heidelberg, pp. 449-458, 2008.

[6] Chen W., Li J., Ma J., Conradi R., Ji J., and Liu C., "A Survey of Software Development with Open Source Components in Chinese Software Industry," *in Proceedings of the International Conference on Software Process Dynamics and Agility*, Springer-Verlag Berlin Heidelberg, pp. 208-220, 2007.

[7] Councill B. and Heineman G., "Definition of a Software Component and its Elements," *Component-Based Software Engineering Putting the Pieces Together*, Addison-Wesley Professional, 2001.

[8] Crnkovic I., Chaudron M., Sentilles S., and Vulgarakis A., "A Classification Framework for Component Models," *in Proceedings of the 7th Conference on Software Engineering and Practice in Sweden*, 2007.

[9] Crnkovic I. and Larsson M., "Introduction," *Building Reliable Component-Based Software Systems*, Artech House, 2002.

[10] Gutierrez P. and Escalante M., "CBSE State of Practice and Experience Survey Preliminary Results," Slides presented at the 2nd Series of CBSEnet Component-Based Software Engineering Workshop, available at http://miro.sztaki.hu/projects/cbsenet/budapest/presentation/Pedro-229CBSE.ppt, 2004.

[11] Herold S., Klus H., Welsch Y., Deiters C., Rausch A., Reussner R., Krogmann K., Koziolek H., Mirandola R., Hummel B., Meisinger M., and Pfaller C., "CoCoME: The Common Component Modeling Example," *The Common Component Modeling Example Comparing Software Component Models*, LNCS 5153, Springer Berlin Heidelberg, pp. 16-53, 2008.

[12] Kotonya G., Sommerville I., and Hall S., "Towards a Classification Model for Component-Based Software Engineering Research," *in Proceedings of the 29th EUROMICRO Conference*, pp. 43-52, 2003.

[13] Kung-Kiu L. and Wang Z., "Software Component Models," *IEEE Transactions on Software Engineering*, IEEE Computer Society Press, pp. 709-724, 2007.

[14] Li J., Conradi R., Slyngstad O., Bunse C., Khan U., Torchiano M., and Morisio M., "An Empirical Study on Off-the-Shelf Component Usage in Industrial Projects," *in Proceedings of the 6th International Conference*, LNCS 3547, pp. 54-68, 2005.

[15] Lüer C. and Hoek A., "Composition Environments for Deployable Software Components," *Technical Report*, Department of Information and Computer Science, University of California, Irvine, 2002.

[16] Pyne R., McNamarah S., Bernard M., Hines D., Lawrence G., and Barton D., "An Evaluation on the State of Component-Based Software Engineering in Jamaica," *in Proceedings of the IEEE Southeast Conference*, 2005.

[17] Rausch A., Reussner R., Mirandola R., and Plášil "Introduction," *The Common Component Modeling Example Comparing Software Component Models*, LNCS 5153, Springer Berlin Heidelberg, pp. 1-3, 2008.

Hazleen Aris is a senior lecturer at the College of Information Technology, Universiti Tenaga Nasional. She obtained her Masters in software engineering from the University of Malaya and a Bachelor of engineering (Hons) in computer engineering from the University of Southampton, UK. She teaches software engineering related subjects to the undergraduate computer science and information technology students at Universiti Tenaga Nasional. Her research interest includes component-oriented software engineering and object-oriented software development.

Siti Salim, is a professor at the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya. She holds PhD degree in computer science from the University of Manchester Institute of Science and Technology (UMIST), United Kingdom, 1998. She supervises PhD and Masters students in the areas of requirements engineering, human computer interaction, computer supported cooperative work, component based software development and e-learning. She also leads and teaches modules at both BSc and MSc levels in software engineering.