

## A Systematic Technology Selection Method for Context-Aware Applications Development

Rozilawati Razali, Abdullah Mohd Zin, Mohd Rosmadi Mokhtar and Seyed Hadi Mirisae  
Centre of Software Technology and Management, Faculty of Information Science and  
Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

**Abstract:** Context-aware applications are mobile applications that use settings to provide relevant information and services to users. Technology selection is the process of determining methods, techniques and tools that could fulfill the development requirements of a particular application. Currently, a number of technology selection methods have been proposed such as The Technology Identification, Evaluation and Selection (TIES), Two-Step Optimization Approach and Multiple Criteria Decision Methods (MCDM). None of these methods however are specifically designed for selecting the suitable technology for developing context-aware applications. This study proposes a systematic technology selection method for context-aware applications that comprises three stages; understanding user requirements, comparing available technologies and selection. A case study involving technology selection for implementing a University Mobile Organizer has been carried out in order to evaluate the feasibility of the proposed approach. The result of the study suggests that the proposed method is feasible and suitable.

**Keywords:** Context-aware, framework, heuristic evaluation, J2ME, JADE, JCAF, spring NET

### INTRODUCTION

Software engineers are provided with an enormous choice of technology for improving software development process and product. It is unacceptable to advocate the efficacy of a new technology without accompanying scientific evidence. The evidence does not only demonstrate the technology's true potential but also ease the selection process required by software engineers during development. The evidence also acts as a means to justify the choices made in the process.

Technology selection is the process of determining methods, techniques and tools that satisfy criteria reflecting a particular target community's requirements (Riddle and Lloyd, 1987). In any engineering discipline, a high quality process normally leads to a high quality product. It is believed that when the right methods, techniques and tools are selected, the quality of the end product can be assured.

Selecting a potential technology to be used in any development is not a straightforward task. Empirical evaluation such as surveys, experiments and case studies provide an ideal mechanism for evaluating software technology. As a single evaluation can never hold all possible conditions, it is reasonable to acquire some evidence of a new technology's effectiveness when used under certain circumstances.

Context-aware applications are a type of mobile applications that use contexts to provide relevant information and services to users. Context-aware applications know the users' environment and execute functions automatically without requiring explicit control or notification.

A number of technologies have been invented for developing context-aware applications such as Java Agent Development Environment (JADE), Java Context-Aware Framework (JCAF), Java 2 platform Micro Edition (J2ME) and Spring.NET. Due to rapid advancement of mobile industry, more context-aware applications development is expected. To date, the mechanism for selecting the suitable technology for context-aware applications development however is almost non-existent. The available selection methods are not specifically designed for context-aware applications. Thus, there is a need for such a method to address this particular issue.

This study presents a study that aimed to tackle the above-mentioned research deficiency. In particular, the study proposed a systematic selection method for selecting the suitable technology for context-aware applications. In order to evaluate its feasibility, the proposed method was validated in a case study.

**Background:** There are a number of technology selection methods that have been proposed such as Technology

Identification, Evaluation and Selection (TIES), Two-Step Optimization Approach and Multiple Criteria Decision Methods (MCDM).

TIES was developed to address the challenge of determining successful resource allocation strategies amongst a large set of possible alternatives (Kirby and Mavris, 2002). TIES identifies a technology from a pool of possible combinations through a technology impact model that translates the combinations to a vector of terms called 'k-factors'. A pre-computed meta-model then relates the vector of k-factors to the system objectives and maps it to a response value. When the number of candidate technology is large, so is the number of possible technology sets. This causes optimisation problem. In cases where the combinatorial space is difficult to search, the use of heuristics approach can aid the situation by providing access to more local optima. The use of Genetic Algorithms (GA) within the TIES process can act as a means of solving the constrained combinatorial optimisation problem.

Utturwar *et al.* (2006) proposed a Two-Step Optimization approach for technology selection as an improvement of TIES. The aim of this study was to obtain an efficient optimisation approach for the technology selection in the aircraft design process. Aircraft designers are normally required to search for the right technology within a set of technology options. When the set of available technology is large, a difficult combinatorial optimisation problem arises that result in significant time and computational expenses. The objective of the approach is to reduce the computational cost of technology selection by decomposing the process into two smaller sub-problems. The approach exploits the structure of the technology compatibility matrix to improve the efficiency of the technology selection process.

Ondrus *et al.* (2005) has proposed a Multi-Actor, Multi-Criteria Approach for Technology Selection when designing mobile information systems. In their proposal, they mentioned that technology selection must be based on Multiple Criteria Decision Methods (MCDM), which has been proven to be useful in decision making process. The application of MCDM in technology selection consists of the following six steps:

- Definition of the problem and its alternative solutions
- Identification of the stakeholders
- Definition of selection criteria
- Selection of the technique of preferences aggregation
- Evaluation of solutions in respect to each selection criterion
- Search for consensual solution among the actors

There are many actors that need to be considered, which can be classified as direct actors (players) and

indirect actors (government). The government sets a legal framework by making and enforcing rules. Players on the other hand comprise technology suppliers, service providers and consumers.

Previous studies on the technology selection methods seem to investigate the factors contributing to this issue individually. Methods such as TIES and Two-Step Optimization focus more on the technical aspects that is how to determine and employ the matrix. MCDM on the other hand emphasises on the selection process itself. The proposed method in this study is intended to look at the issue from both technical and process perspectives. Moreover so far, the methods proposed by previous studies were meant for domains other than context-aware applications.

## MATERIALS AND METHODS

This study presents a technology selection method that comprises three steps as shown in Fig. 1. The first step in technology selection process is to obtain a clear understanding of user requirements. This is very important for context-aware applications as they interact with users directly and automatically. As the method is meant for context-aware applications, the requirements must satisfy the specific characteristics of such applications. User requirements can be massive therefore some kind of prioritisation technique has to be employed. Each of the identified requirements is given a weightage that indicates the importance of the requirement from users' points of view. The weightage weighs the requirements or functions of concern to total 100% (or one by value). For example in Table 1, there are four requirements that have been identified for one context-aware application where Function-4 carries the highest weightage (0.35 or 35%). The schema is then used to

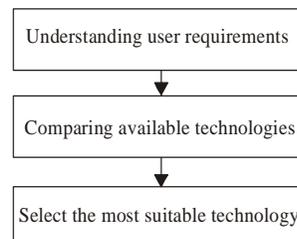


Fig. 1: A technology selection method for context-aware application

Table 1: An example of functions and its weightage

Function	Weightage
Function-1	0.25
Function-2	0.2
Function-3	0.2
Function-4	0.35

Table 2: An example of functionalities provided by each technology

	Technology- A	Technology- B	Technology- C
Function-1	Yes	No	Yes
Function-2	No	Yes	Partially
Function-3	Yes	Yes	Yes
Function-4	Yes	No	No

compare possible technology for context-aware applications development.

The second step is to determine the compliance level of each technology towards the requirements. There are three possible levels: Yes; No; and Partial. Table 2 illustrates an example of how much Technology-A, Technology-B and Technology-C satisfy Function-1 to Function-4.

Values in Table 1 can be represented as a 1 x m matrix W. Values in Table 2 can also be represented as an m x n matrix F, where “Yes” can be replaced by 1, “No” by 0 and “Partially” by 0.5. The capability factor (C-factor) for each technology with respect to the requirements can be calculated as follows:

$$C = W \times F$$

Based on the examples in Table 1 and 2, the matrix W is:  $W = (0.25 \ 0.2 \ 0.2 \ 0.35)$  whereas the matrix F is:

$$F = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0.5 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Therefore, the C-factor can be determined as:

$$C = (0.25 \ 0.2 \ 0.2 \ 0.35) \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0.5 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} = (0.8 \ 0.4 \ 0.55)$$

The C-factor for Technology-A, Technology-B and Technology-C are 0.8, 0.4 and 0.55, respectively. As Technology-A has the highest C-factor it therefore should be selected as the best possible option for the development. This is the third step in the process.

## RESULTS

In order to demonstrate the feasibility of the proposed method, a case study was employed. The case study involved developing a mobile organizer. Mobile organizer is a context-aware application that acts as an assistant for daily practices such as arranging meetings or providing required information regarding user activities. A University Mobile Organizer (UMO) is a special type of mobile organizer that can be used by university students to organise their indoor and outdoor activities in the university environment. One example of UMO is a mobile learning organiser for University of Birmingham students (Corlett *et al.*, 2005). The application has several features

such as a timetable, lecture notes, course manager and important dates for students. This application is supported by the university’s wireless network. Similarly, the case study aimed to develop an UMO by firstly determining the suitable technology by using the proposed method. The following paragraphs describe the steps in detail.

**Step 1 understanding user requirements:** The requirements for the UMO were obtained by conducting a survey at one public university (Mirisae and Zin, 2008a). Twenty-nine females and twenty-one male students participated in the survey. They were postgraduate and undergraduate students with the ratio 1:3. This was approximately the same as their distribution rate in the campus.

Thirty-eight students (76%) had hand-held devices such as notebooks and PDAs. While most of the notebooks operated under Windows operating system, PDAs were operating under various types of operating systems such as Symbian and Windows Mobile. Forty-one students (82%) were able to connect to the university’s wireless network. This implies that wireless network in the campus was not a problem and thus, they would be able to use the proposed mobile organizers.

The survey found that twelve students (24%) strongly agreed that a UMO could help them in accessing their required online information more easily. Forty-seven students (94%) preferred to use the application to assist them in their university activities. The common concerns raised by the students (80%) were security and privacy such as disclosing personal information and accessing online information through the wireless network provided by the university.

In general, the UMO features needed by students can be classified into two types. Thirty-eight students (76%) indicated the system should support their learning process by giving information concerning their courses such as class schedule; ability to send and receive messages to/from lecturers; online submission of assignments and downloading course material. The access to these features could be made from any location. The second type of features that was mentioned by thirty-three students (66%) were related to context such as information about bus schedule, shops or other points of interest close to their current locations.

The results of the survey indicated that the main requirements of UMO are:

- Support many different types of devices that operate under various types of operating systems.
- The system must allow access to be made from remote locations.
- The system must be able to take into consideration the context that can change dynamically as users move from one location to another.

- The system must give serious consideration on issues related to security and privacy.

**Step 2 comparing available technologies:** The case study identified four possible technologies to be used for developing the UMO. They were Java Agent Development Environment (JADE), Java Context-Aware Framework (JCAF), Java 2 platform Micro Edition (J2ME) and Spring.NET.

Bellifemine *et al.* (1999) defines JADE as software to develop agent applications in compliance with the Foundation for Intelligent Physical Agents (FIPA). JADE provides a programming interface to simplify the registration of agent services with one or more domains. It also provides a Graphical User Interface (GUI) to manage agents and agent platforms from an agent. JADE provides a library for the automatic registration of agents and managing different agents. Therefore, an agent can use other agents in its context and manage them properly. JADE can use Java components to run in different operating systems on mobile devices. The main shortage of JADE is that no security policy is considered in order to secure data in the transferring phase. Researchers have used JADE in developing agent-based context-aware frameworks (Ilarri *et al.*, 2010).

JCAF is a framework for designing and implementing context-aware applications (Bardram, 2005a). JCAF's simplicity is in using Java language programming and Java Remote Method Invocation (RMI). Its main feature for developing context-aware applications is the ability to dynamically download and run codes. JCAF is divided into two main parts: a Context-Awareness Runtime Infrastructure and a Context-Awareness Programming Framework. The design of Context-Awareness Runtime Infrastructure core includes four principles (Bardram, 2005b): A service for linking services with different contexts to build meaningful information for the application; an active infrastructure that takes responsibility for interaction with the environment; the ability of extension since the context types are changing based on situation; and adding new context services which are required for new context sensors. JCAF has been used for developing of context-aware frameworks (Bardram, 2009).

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems" (Sun Micro System developer Works group, 2007). J2ME is recognised as a link to the next generation of mobile applications since it has a larger community footprint. A large number of devices support J2ME for their application development environment (Christensen, 2009). J2ME uses configurations and

profiles to customise the Java Runtime Environment (JRE). The most significant benefit of J2ME is that it allows developers to write portable codes, which enables them to change context types dynamically. Other advantages of J2ME are (Padhiyar, 2007):

- The class verification is done in loading time, so running any dangerous codes for mobile devices is prevented.
- J2ME's monitoring system safeguards runtime errors. This feature is useful for monitoring new codes, which are added to context-aware applications in runtime.
- Network and data security can be guaranteed by establishing point-to-point secure connections.

Spring.NET is a new technology for implementing enterprise applications in the .NET framework. It might be useful as a solution for implementing context-aware applications. One of the advantages of Spring.NET is that developers use the required parts without loading or knowing about the whole library or class (Pollack *et al.*, 2008, 2011). Spring.NET provides features to connect remote objects and dynamically changes its code and behavior to adapt to different situations. Spring.NET remoting is a component of Spring.Net technology that allows developers to create and export services to client applications. There are two methods for exporting services through Spring.NET remoting. Firstly, a Client Activated Object (CAO) is created by user request and destroyed when the user does not refer to it anymore. It is created to one user only (MSDN, 2008a). Secondly, a Server Activated Object (SAO) is created when the client makes a method call from the object and the server directly controls lifetimes (MSDN, 2008b). These methods support privacy and security for context-aware applications. Furthermore, Spring.NET's various possibilities can reduce the complexity of context-aware applications development and are useful for enterprise context-aware applications. Supporting different programming languages is a specific Spring.NET's feature which can reduce time and cost of developing. The significant drawback of Spring.NET technology is its incompatibility with different operating systems. These applications are only working on Microsoft operating systems, besides Java based technologies.

The comparison of features provided by these technologies is given in Table 3, which has been described in detail by Mirisae and Zin (2008b).

**Step 3 select the most suitable technology:** Based on the results of the survey, the weightage for each function was determined as shown in Table 4. F3, F4, F5 and F6 were given 0.2 as they are the main requirements of UMO,

Table 3: A summary of features for different technologies

Specifications	Spring.NET	JADE	J2ME	JCAF
F1: Support remote connections protocols such as HTTP	Yes	Yes	Yes	Yes
F2: Can support different context programming languages	Yes	No	No	No
F3: Compatible with different operating systems	No	Yes	Yes	Yes
F4: Changing context type dynamically	Yes	No	Yes	Yes
F5: Offer suitable methods for security	Yes	No	Yes	No
F6: Offer suitable policy for privacy	Yes	No	No	No

Table 4: Functions and weightages

Specifications	Weightage
F1	0.1
F2	0.1
F3	0.2
F4	0.2
F5	0.2
F6	0.2

$$W = (0.1 \ 0.1 \ 0.2 \ 0.2 \ 0.2 \ 0.2)$$

mentioned in earlier section. F1 and F2 were given 0.1 as they are less important requirements, which exist but were not discovered in the survey (Step 1).

Based on Table 3, the functionality matrix F can be stated as:

$$F = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Therefore, the C-factor can be calculated as:

$$C = (0.1 \ 0.1 \ 0.2 \ 0.2 \ 0.2 \ 0.2) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = (0.8 \ 0.3 \ 0.7 \ 0.5)$$

Table 5: Heuristics evaluation by experts

Heuristics	Comments from experts
Approaches to flexibility and extensibility	Inversion of control in Spring.NET automatically creates objects and inserts the dependencies in creation process. This reduces the complexity of adding new components.
Separation of concerns	Spring.NET supports this very well.
Scalability	Spring.NET has no limitation for running the application with increasing users.
Historical data and reusability	Spring.NET does not support reusability very well but can be improved if the architecture is refined.
Support during development, debugging and deployment	The graphical Integrated Development Environment (IDE) improves the development process. Testing and code coverage analysis are also supported by .NET framework.
User explanations	More dynamic information in different parts of the application is required.
Security and privacy	Security is at an acceptable level. Privacy is guaranteed by Spring.NET methods and description of services as a private or public service is done by that.
Reliability and robustness	Robustness is considered. When a local server is disconnected, user can still get information from the main server.
Match between context-aware system and the real world	Reasonably supported.
Reconfiguration and maintenance	Not really supported but can be improved by having a remote configuration manager component that configures services on servers and controls updated configuration files for users.
Manual override	Manual configuration gives options to users to determine its location manually or automatically.

The value of C seems to suggest Spring.NET as the best technology to be used for implementing UMO.

### DISCUSSION

To confirm the suitability of using Spring.NET in developing a context-aware application, a UMO was developed (Mirisae and Zin, 2009) and heuristics evaluation (Nielsen and Mack, 1994) by experts was conducted on it. The development environment used for the UMO development was Visual Studio.Net. The programming language was C# and the database was SQL Server 2005. The UMO ran on Microsoft operating systems. Windows Server 2003 provided a secure environment for server applications and files management. Spring.NET was used for context-aware aspects such as transferring services or context reasoning.

The heuristic evaluation was done by six experts from three different areas: communication, mobile computing and software development. This combination increases the chances of finding problems from various aspects of the application. The evaluation was based on eleven standard heuristics (Yoosoo *et al.*, 2007). Table 5 lists the heuristics and the respective comments received from the experts.

According to the evaluation done by the experts, most heuristics are supported by the .NET based technologies. These findings indicate that Spring.NET with integration

of other. NET-based technologies can be a suitable technology for developing and supporting context-aware applications. This result aligns with the C-factor calculated earlier. This also implies that the proposed technology selection method is valid, feasible and suitable to be used for context-aware applications.

## CONCLUSION

This study has presented a systematic technology selection method for context-aware applications, which comprises three stages: understanding user requirements, comparing available technologies and selecting the suitable technology. In order to test the validity and suitability of the proposed method, a context-aware application, namely University Mobile Organizer (UMO) was developed based on Spring.NET technology. Spring.NET was chosen because it was suggested by the proposed method as the most suitable technology for UMO development. To confirm the selection made, the study conducted heuristics evaluation on the developed UMO. The evaluation was done by six experts from communication, mobile computing and software development fields. The results from the evaluation support the selection made. This indicates that the proposed method is feasible and suitable to be used for selecting the best technology for context-aware applications. To enhance its accuracy, this method needs to be refined further and tested in different context-aware applications in future.

## REFERENCES

- Bardram, J.E., 2005a. Design, Implementation and Evaluation of the Java Context Awareness Framework (JCAF). Center for Pervasive Healthcare, pp: 1-14.
- Bardram, J.E., 2005b. The Java Context Awareness Framework (JCAF)-A Service Infrastructure and Programming Framework for Context-Aware Applications. Centre for Pervasive Computing, pp: 1-18.
- Bardram, J.E., 2009. Activity-based computing for medical work in hospitals. ACM Trans. Comput. Hum. Interact., 16(2): 1-36.
- Bellifemine, F., A. Poggi and G. Rimassa, 1999. JADE-A FIPA-compliant agent framework. in the Practical Applications of Intelligent Agents. 1999: Citeseer.
- Christensen, J.H., 2009. Using RESTful web-services and cloud computing to create next generation mobile applications. 24th ACM SIGPLAN Conference Companion on Object-oriented Programming Systems Languages and Applications, ACM, Orlando, Florida, USA, pp: 627-634.
- Corlett, D., M. Sharples, S. Bull and T. Chan, 2005. Evaluation of a mobile learning organizer for university students. J. Comput. Assist. Learn., 21(3): 162-170.
- Ilari, S., E. Mena, A. Illarramendi and G. Marcos, 2010. A location-aware system for monitoring sport events. 8th International Conference on Advances in Mobile Computing and Multimedia. Paris, France, ACM, pp: 297-304.
- Kirby, M.R. and D.N. Mavris, 2002. An Approach for the Intelligent Assessment of Future Technology Portfolios, AIAA Paper 2002-0515, January 2002.
- Mirisaee, S.H. and A.M. Zin, 2008a. Determining IT students Problems in an University Environment and Proposing a Solution by a Wireless Organizer. International Symposium on Information Technology.. Kuala Lumpur, Malaysia, IEEE.
- Mirisaee, S.H. and A.M. Zin, 2008b. Evaluating spring. Net technology for developing context-aware applications in comparison of JADE and JCAF. 4th Malaysian Software Engineering Conference, R. Mohamad, et al., Editors. 2008b, Penerbit UMT, Malaysia, pp: 338-341.
- Mirisaee, S.H. and A.M. Zin, 2009. A framework for context-aware university mobile organizer. International Conference on Future Computer and Communication, 2009. ICFCC 2009. Kuala Lumpur, Malaysia, IEEE.
- MSDN Server Activation, 2008a. Retrieved from: [http://msdn2.microsoft.com/en-us/library/y0h540a7\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/y0h540a7(VS.71).aspx).
- MSDN Client activation, 2008b. Retrieved from: [http://msdn.microsoft.com/en-us/library/w93betdk\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/w93betdk(VS.85).aspx).
- Nielsen, J. and R.L. Mack, 1994. Usability Inspection methods. Wiley, New York, pp: 413.
- Ondrus, J., B. Tung and P. Yves, 2005. A Multi-actor, multi-criteria approach for technology selection when designing mobile information systems. IFIP TC8 Working Conference on Mobile Information Systems (MOBIS'05), 5-6 Dec 2005, Leeds, UK.
- Padhiyar, J., 2007. Overview Advantages of J2ME. Retrieved from: <http://ezinearticles.com/?Overview-Advantages-of-J2ME&id=890306>
- Pollack, M., R. Evans, A. Seovic, F. Spinazzi, R. Harrop, G. Caprio and C. Rim, 2008. The Spring. NET Framework Reference Documentation. Retrieved from: <http://www.springframework.net/docs/1.1.1/reference/html/index.html>.
- Pollack, M., R. Evans, A. Seovic, B. Baia, E. Eichinger, F. Spinazzi, R. Harrop, G. Caprio, R. Bartelink, C. Rim, E. Mazor and S. Bohlen, 2011. The Spring. NET Framework. Retrieved from: <http://www.springframework.net/doc-latest/reference/html/index.html>.

- Riddle, W.E. and G.W. Lloyd, 1987. Technology selection: An educational approach. *IEEE Trans. Software Eng.*, 13(11)
- Sun Micro System developer Works group, 2007. J2ME: Step by step. Retrieved from: [www.digilife.be/quickreferences/pt/j2me%20step%20by%20step.pdf](http://www.digilife.be/quickreferences/pt/j2me%20step%20by%20step.pdf)
- Utturwar, A., S. Rallabhandi, D. Delaurentis and D. Mavris, 2006. A two-step optimization approach for technology selection, engineering optimization. *Eng. Optim.*, 38(8): 889-908.
- Yoosoo, O., A. Schmidt and W. Woo, 2007. Designing, developing and evaluation context-aware systems. *International Conference on Multimedia and Ubiquitous Engineering IEEE Computer Society*, pp: 1158-1163.