A CMMI Based Configuration Management Framework to Manage the Quality of Service Based Applications

Sajid Ibrahim Hashmi, Stephen Lane, Dimka Karastoyanova[1], and Ita Richardson

Lero – The Irish Software Engineering Research Centre
University of Limerick, Ireland
{sajid.hashmi, stephen.lane, ita.richardson}@lero.ie
IAAS, University of Stuttgart, Germany[1]
dimka.karastoyanova@iaas.uni-stuttgart.de[1]

Abstract

Service Based Applications (SBAs) have highlighted new challenges related to Configuration Management (CM). This is an important process for the assurance of end to end quality in software systems. As far as the quality of SBAs is concerned, configuration management remains an issue because of the loosely coupled and adaptive nature of the corresponding applications. A smart configuration management approach will allow organizations to make their IT resources more reliable and to utilize them to their maximum. In this paper, we propose a service-based configuration management framework based on SEI CMMI-SVC which contributes to the S-Cube life cycle. Implementing this approach will allow organizations to effectively manage the configurations of their SBAs.

Keywords

Service Oriented Architecture, SBAs (Service Based Applications), CM (Configuration Management), SC (Software Configuration), Quality Assurance, CMMI – SVC (Capability Maturity Model Integration for Services)

1 Introduction

Today's computer world consists of applications which are scattered across different networks and require special effort in terms of integration. For their smooth operation, developers of such applications need to pay special attention to configurations as 60% of service impacts are due to configuration problems [1]. Organizations have business processes in place in order to meet their objectives – for example, sales, administration, and financial departments work together in a "Sales" process. Each of the units involved in an organization needs one or more services (e.g. application software or utilities). These services run on IT infrastructure which includes both hardware and software, therefore it must be managed accordingly to meet organizational objectives [3]. Proper management of IT infrastructure will ensure that the required services by business processes are available.
Configuration Management (CM) is a Software Quality Assurance (SQA) process for managing different configurations of configurable software items (Galin, 2003). In addition, it is part of the IT

infrastructure which consists of procedures, policies, and documentation. Many items change during a software product's lifetime and it is important to keep track of these changes. Customers may have different software versions so it is important to know which version each customer is using in order to support them effectively. This will facilitate customer support as for queries it may be necessary to easily access various version of source code, design documents or support documentation. Issues related to poor Configuration Management (CM) include system related failures, failure of key services, deficiency in performance and reduction in employee productivity, all of which consequently can cause serious business impact. In short, CM is a quality enabling process which provides a logical view of services by identifying, maintaining, and verifying the versions as well as the corresponding configuration items [2].

In service-oriented environments the heterogeneity of resources is dealt with by providing any kind of functionality or resource as a service with a stable interface. However this does not completely remove the need for configuration of the resources, which has to be performed by any service provider of an SBA. Software development is a dynamic process where systems are constantly refined and modified [20]. Consequently, as the system evolves, an efficient CM process becomes increasingly important. Software systems are developed individually, but these systems are integrated to gain the benefits of Service Oriented Architecture. This integration to achieve exchange of information gives rise to different management issues. This is because complexity of data exchange increases as the number of services increase. Additionally, CM process activities will getting increasingly complex as the number of services increase, hence causing the CM process itself to be modified regularly.

Software quality assurance is about identifying the right things to implement and test, and allocating and managing resources in a way that minimizes risks when applications and services are deployed [4]. There are two types of quality assurance activities [21]: constructive and analytic quality assurance. The purpose of the constructive quality assurance is to prevent fault injection when artifacts are being created. Analytic quality assurance deals with cleaning artifacts after they have been constructed. In this research, our aim is to support constructive quality assurance, i.e. to prevent defect injection at design time.

In this paper, we present the development of an initial CM framework that can contribute to the end-to-end quality assurance of SBAs. Effective CM will support the effective management of SBA configuration, and this should help to assure their quality. In terms of end to end quality, a CM process would allow developers more accurately develop and update the correct versions of services. SBAs or other service consumers would also benefit from CM as they would get to know when services get updated, allowing them to update accordingly. The remainder of the paper is organized as follows: the remainder of Section 1 describes background information, and Section 2 describes research methodology, the framework, and the example scenario to validate the applicability of the proposed approach, and finally we sum up our conclusions in section 3 .

1.1 S-Cube

S-Cube project is funded by European Community's 7th Framework Programme [10]. Its objective is to create an integrated European research community in the area of software and service engineering. It is based on an ideology that the engineering and management of SBAs is quite different to traditional software applications as they are built by combining different services which may be provided by third parties with whom there should be a service level agreement.

A reference lifecycle for SBAs has been developed by S-Cube project researchers (see Figure 1). It is composed of two cycles. The evolution cycle depicts classical application design while the adaptation cycle reflects the adaptation of SBAs. SBAs need to accommodate many changes at run time and this two cycle approach provides a balance between the design and runtime operation. The Operation and Management phase, where CM resides belongs to both phases. Therefore, it must be efficient and precise enough to meet the transition needs of the entire life cycle. By further defining the CM process within Operation and Management, the research presented in this paper aims to strengthen the S-Cube life cycle.

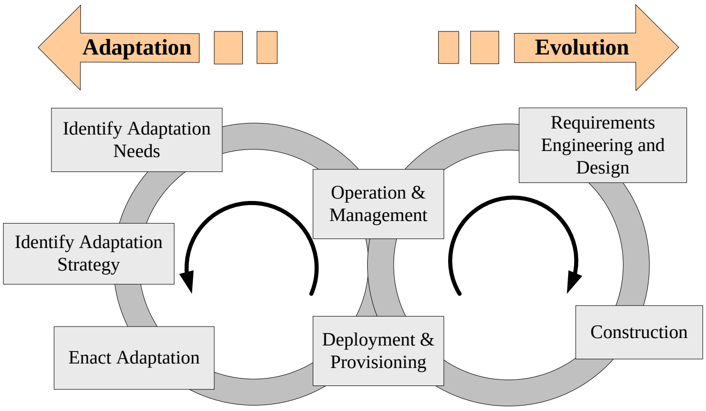

Figure 1: S-Cube Reference Life Cycle [10]

1.2 CMMI-SVC

Capability Maturity Model Integration (CMMI) [5] models are a collection of best practices that help organizations to improve their processes. CMMI - SVC [6] is a CMMI assemblage that covers the activities designed to manage, establish, and deliver services. It has been designed for service industry as a process improvement framework and its goals and practices are relevant to any organization concerned with the delivery of service. CMMI – SVC includes 25 process areas sub-divided into 4 process categories. We used the expert judgment technique [11] to identify process areas and practices which can support CM practices for service based applications.

1.3 Background

In component based development (CBD), software applications can be made up from several standalone components [12]. In SBAs services work as components, and for a good CM, each component included in each application release should be recorded. If the versions of the components are changed then the overall application version should change. This facilitates the quality assurance of the entire software system. In traditional software systems, CM can be achieved successfully if a suitable process guideline or standard is followed. Examples are: IEEE 828:2005, the IEEE standard for software configuration management plans [13] or Leon's guide to software configuration management [14]. Our starting point for the development of the service-based CM framework is the CM process as outlined by Galin [7]. We make use of it because it is comprised of set of configuration activities and their associated action items (see Figure 2). In addition, we map it with the relevant process areas and practices in CMMI-SVC because Galin's model alone did not fullfill the requirements of service based applications. We chose these two models because of their wide use for process management and quality assurance

2 Research Methodology

To develop a service-based CM framework, we make use of a traditional software engineering CM process and supplement it with applicable practices from CMMI-SVC. Galin [7] has defined four levels of CM activities which further contain sets of action items. For each action item, we identified those CMMI-SVC process area(s) and subsequent practice(s) which could support the implementation of that actions item. This one by one mapping of CM supporting action items with relevant CMMI-SVC practices has allowed us to identify an initial service-based CM framework. We then illustrate the implementation of the framework through an example scenarion.

2.1 Initial CM Framework

Galin's four activities are: Software Change Control, Release of Configuration Items (CI) and Software Configuration Versions, Provision of CM Information Services, and Verification of Compliance to SCM Procedures. In our mapping, a process area or practice may be used multiple times to implement different CM activities. Table 1 illustrates the first level of our framework - it displays the mapping between the action items for Software Change Control and appropriate practices from CMMI-SVC and CMMI.

Table 1. Description of CM Framework

Action Items (from Galin)	Relevant CMMI – SVC Process Areas	Corresponding CMMI Practices
Grant approval to carry out changes	• Service System Transition	• Analyze Service System Transition Needs • Prepare Stakeholders for Changes
	• Strategic Service Management	• Gather and Analyse Relevant Data
	• Configuration Management	• Establish change management system
	• Requirements Management	• Manage Requirements Changes • Identify Inconsistencies between Project and Requirements
Control the changes and assure quality of approved changes	• Process and Product Quality Assurance	• Objectively Evaluate Processes • Objectively Evaluate Work Products • Establish Records
Document the approved changes	• Configuration Management	• Establish CM Records • Perform Configuration Audits
Mechanism to prevent simultaneous changes in the same SC item	• Configuration Management	• Track Change Requests • Control Configuration Items

The first CM activity in the framework is Software Change Control. This is an important activity which ensures changes to software systems are carried out with the appropriate levels of governance. This prevents inappropriate or unsafe changes from being made without approval, and becomes particularly important in SOC where changes to services may affect many downstream SBAs. In order to implement the action items, suitable practices were taken from the CM, Requirements Engineering, and Process and Product Quality Assurance process areas of CMMI-SVC and CMMI. The CM and Requirements Engineering process areas provided practices for the steps required to implement software change control, while the Process and Product Quality Assurance process area provided practices for quality assurance during this process.

The second activity in the framework is Release of Software Configuration Items and Software Configuration Versions. When new software versions are released it is important to record version and installation details. This information assists with trouble shooting and diagnosing software errors. With regard to services, the recording of installation sites is not usually an issue as they are usually installed in one location with multiple applications accessing the same services. The release of software configuration items and software versions have differing implications depending on whether services or SBAs are being considered. When new versions of services are released it is important to have access to details of previous versions in the event that they are required. An example would be an incompatibility issue with a service consumer. When SBAs are considered a new application version may be released by adding services or removing services from an existing SBA. Similarly an SBA may require a new version if its component services are updated. In both of these cases configuration details and document version releases should be recorded. Documentation and source code for each release is an important resource for support and quality assurance activities. This activity can be achieved in SOC using practices from CMMI-SVC activities such as CM, Project Monitoring & Control, and Process and Product Quality Assurance.

The third activity in the framework is Provision of Software Configuration Information Services, it ensures that information about status of changes, versions, and documentation is maintained. Whereas, the fourth activity Verification of Compliance to Software Configuration Procedures deals with verifying compliance to SCM procedures. Our identification of the practices has allowed us to modify the S-Cube life-cycle as shown in Figure 2.

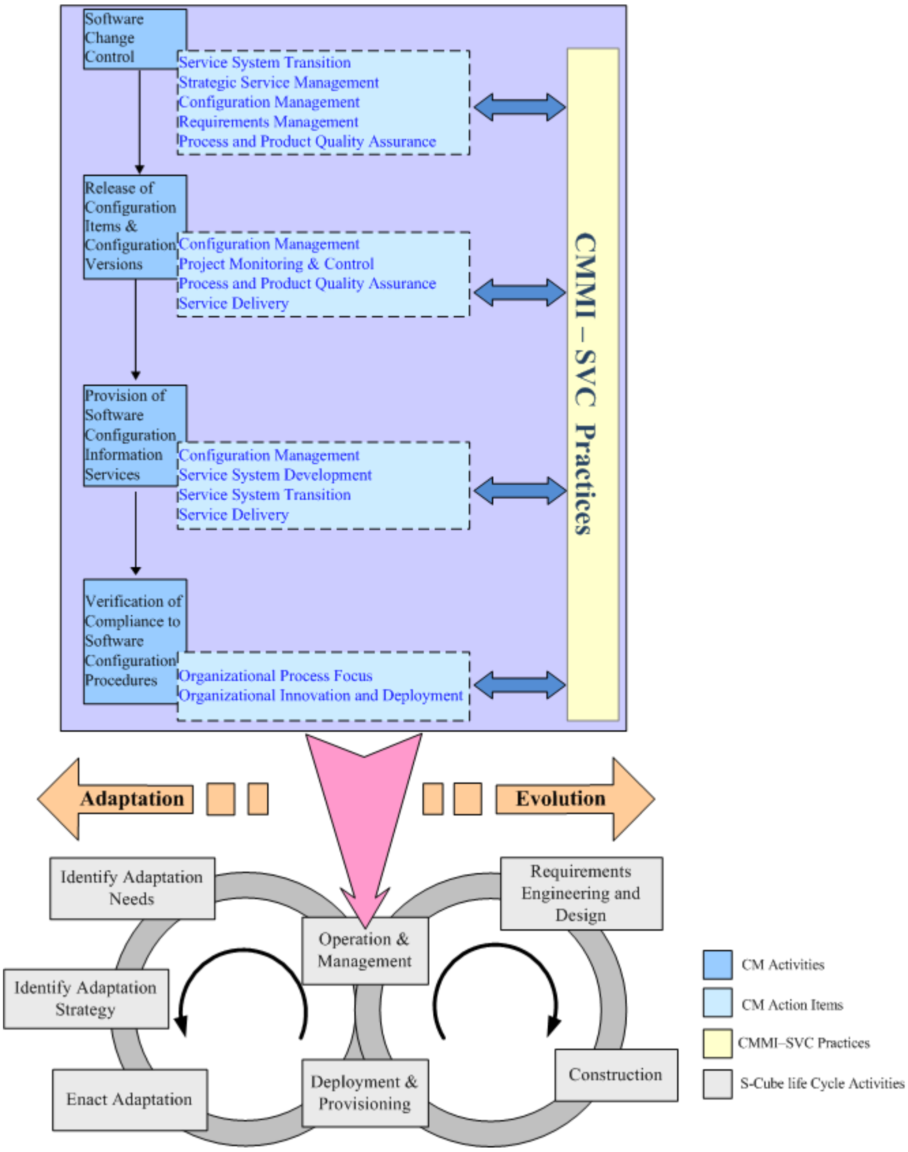

Figure 2: S-Cube Lifecycle with Detailed Operations and Management Phase

2.2 Example Scenario

An example scenario has been designed in the S-Cube project. This is a complex and geographically distributed supply chain in the automotive sector which has been offered by researchers of the companies 360Fresh and IBM [9]. We use this case study to illustrate the possible implementation of our proposed framework. We determined business goals and domain assumptions for the purpose of this illustration. Figure 3 illustrates the global business scope of the service network in the case study. It highlights the main actors and the interactions, concerning both material and information flow, that occurs between them.

The service network consists of multiple warehouses, scattered across different geographical locations where finished products are stored from the manufacturing factory. If management want to reorganize the current set up (reasons might be due to changing demand patterns or the termination of a lease for a number of existing warehouses), they need to consider their business challenges.

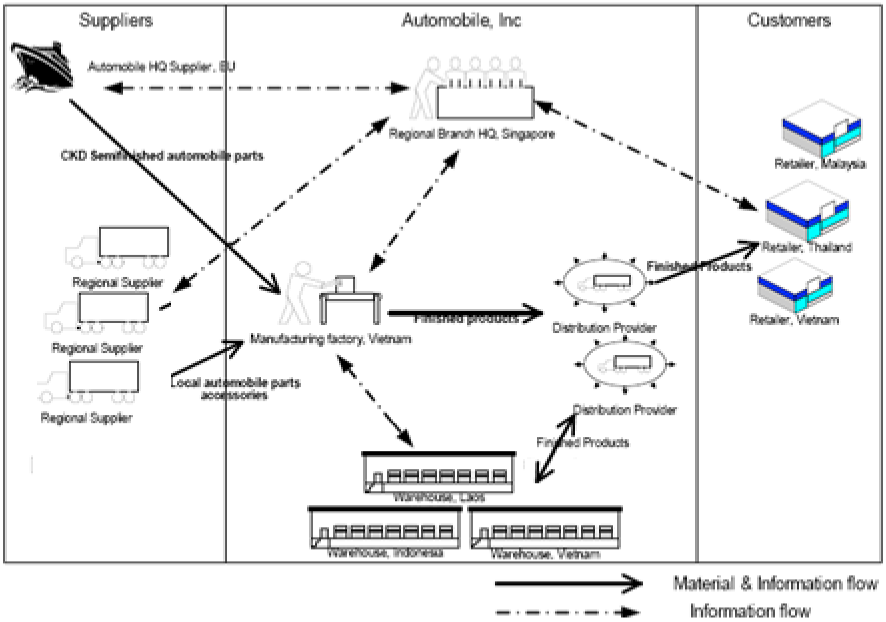

Figure 3: The Business Scope of the Service Network

Reconfiguring a distribution network may require subsequent changes, such as a new flow pattern of goods throughout the network or a change in production levels. They need to consider how to select their new warehouse locations in order to meet changes in demand patterns. The overall purpose of the network is that merchandize is produced and distributed at right quantities, to the right locations, and at the right time, in order to minimize the system wide costs while satisfying service level requirements. For the sake of simplicity we show the warehouses (W_1-W_3) in the network (Figure 4).

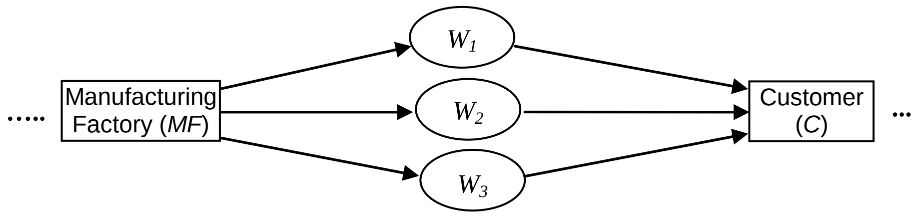

Figure 4: A Simplified View of Service Network

Our objective is to manage the distribution network and reconfigure it based on the optimal strategy when different warehouses are merged into one. The distribution strategy must be able to manage the flow of products from the suppliers through the warehouse to the market areas without interruption. We have three ways to route the finished product to the customer.

$$MF \rightarrow W_1 \rightarrow C$$
$$MF \rightarrow W_2 \rightarrow C$$
$$MF \rightarrow W_3 \rightarrow C$$

Several criteria can be considered to make this decision such as cost, overhead, and distance. However, in a complex logistic network, it is too hard to obtain the optimal path of the network [15] as there are lots of issues involved. For reconfiguration, the supplier evaluation indexes were first presented by Dickson [16] and Weber [17], and then this index research was expanded [18]. Considering these evaluations and problem indexes allowed us to align them as shown in Figure 5.
The framework we propose in section 2.1 can benefit us in order to address issues associated with fusion of multiple warehouses into a single one. In Table 2, we identified a set of CMMI practices

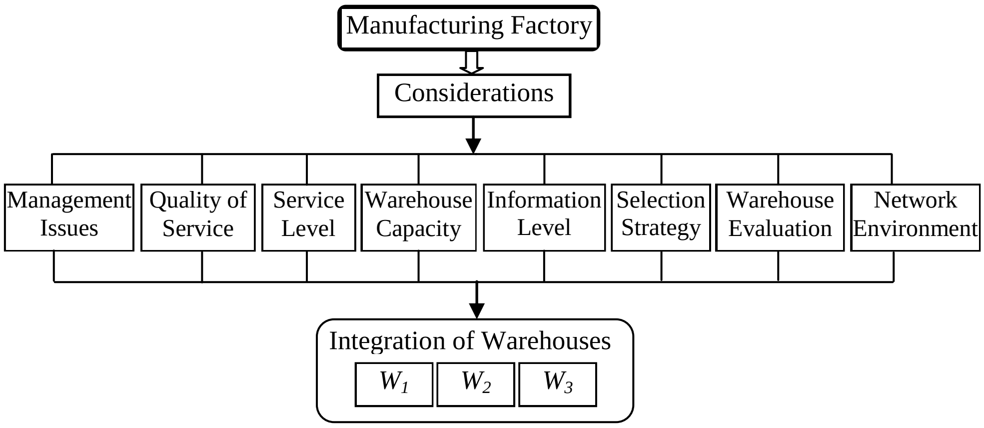

Figure 5: Considerations in Integration of Warehouses

Which can help us to support the CM process, i.e. transition from a multiple into a single warehouse. We may select a set of practices depending on the situation. Thus, our framework has been useful in upporting the business change required.

Table 2: The Service-based CM Framework Supporting Network Modification

Issues	CMMI - SVC Practices	Description
Management Issues	• Analyze Issues • Establish change management system • Prepare stakeholders for Changes	It supports and addresses issues associated with integration
Quality of Service	• Objectively evaluate processes • Objectively Evaluate work products • Conduct Progress Reviews • Select Improvements for Deployment	It supports quality of services by evaluating processes, work products, progress, and improvement deployments
Service Level	• Analyze Service System Transition Needs • Ensure Interface Compatibility • Validate the Service System	Levels of services are addressed by these practices
Warehouse Capacity	• Gather and analyze relevant data • Prepare for service system operations	It addresses assessment of warehousing
Information Level	• Receive and Process Service Requests • Identify configuration Items • Validate the Service System	Information level issues are addresses by related practices
Selection Strategy	• Establish CM Records	Selection strategy analysis is facilitated by CM records
Warehouse Evaluation	• Objectively evaluate processes • Establish Records	Warehouse evaluation is addressed by evaluating current processes and Quality records
Network Environment	• Prepare for Service System Operations • Establish Service Delivery Approach • Deploy Service System Components	Network environment can be readjusted by these practices

3. Conclusions

Services have made the world more connected - allowing producers, consumers, and other human resources to communicate frequently across the globe. The service industry is a significant driver for the growth of worldwide economy. Therefore, guidance on improving service management development can serve as a key contributor to the customer satisfaction, performance, and profitability of the business. In this research, we have proposed a service-based CM framework to manage the configuration of service based applications. The development of the framework is supported by a case which depicts the effectiveness of the approach. A special case with Service Oriented Architecture is that the customer does not see the change of services as long as Service Level Agreements are met. Yet, this is not how it is currently carried out, and therefore remains a future research issue for us. Another issue is that sometimes the providers of services in an SBA do not agree with the SBA provider and this may only be discovered dynamically during execution. We intend to use configuration information for the purpose of audit and for ensuring compliance between them.

4. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube). It was supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero – the Irish Software Engineering Research Centre (www.lero.ie), and a Higher Education Authority grant PRTLI 4 to the Lero Graduate School in Software Engineering.

5. Literature

[1] Enterprise Management Associates. "Configuresoft, Inc.'s Enterprise Configuration Manager: A Continuous Compliance Approach to Configuration Management," 2006.
[2] Configuration Management Best-Practice Recommendations, Sun Services White Paper, May 2007
[3] The ITIL Toolkit – The ITIL Guide, ITIL - Office of Government Commerce, The Stationary Office, London, United Kingdom, Typical Page. [http://www.itil-toolkit.com/itil-guide.htm]
[4] Software Quality Management for SOA: Enterprise quality managers take the helm, white paper, Published by Hewlett-Packard
[5] Capability Maturity Model Integration, Technical Report CMU/SEI-2009-TR-001, ESC-TR-2009-001
[6] Capability Maturity Model Integration for Services version 1.2, Technical Report CMU/SEI-2009-TR-001, ESC-TR-2009-001
[7] Software Quality Assurance: From Theory to Implementation, Daniel Galin, Addison Wesley September 21, 2003
[8] Software Services and System Network, www.**s**-cube-network.eu
[9] J. Sairamesh, S. Zeng, B. J. Steele, and M. A. Cohen, "Dynamic service networks: Case study on supply-chain collaboration for early detection and recovery through information services," technical report available from S-Cube on request.
[10]S-Cube deliverables." [Online]. Available: http://www.s-cube-network.eu/results/deliverables
[11] A Guide to the Project Management Body of Knowledge (PMBOK Guide), Project Management Institute; 2000 ed edition
[12] Business component Factory, P. Herzum, O. Sims J. Wiley & Sons Inc., 2000.
[13] IEEE Std 828-2005 - IEEE Standard for Software Configuration Management Plans, IEEE Computer Society, 2005.
[14] A guide to Software Configuration Management, A.Leon, Artech House, Boston, MA, 1999.
[15] Gao Yi; Lin Xinda; Wang Dong;" A computer based strategy design for automobile spare part logistics network optimization", Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for
[16] G. W. Dickson, "An Analysis of Vender Selection System and Decisions," Journal of Purchasing, vo1.2, pp.5-17, September 1966.
[17] C. A. Weber, 1. R. Current, and W. C. Benton, "Vendor Selection Criteria and Methods," European Journal of Operational Research, vo1.50, pp.2-18, January 1991.
[18] E. L. Wilson, "The Relative Importance of Supplier Selection Criteria: A Review And Update," International Journal of Purchasing and Materials Management, vo1.25, pp.35-41, May 1994
[19] Wanshan Wang, Liang Tang, Shuang Liang, Junming Hou; "A Study on CBR-Based Supplier Selection under Networked Manufacturing", Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on
[20] Nguyen, T.N.; Munson, E.V.; Boyland, J.T.;" An infrastructure for development of object-oriented, multi-level configuration management services", Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on, Page(s): 215 – 224
[21] Omar Alshathry, Helge Janicke, Hussein Zedan,, Abdullah AlHussein, "QUANTITATIVE QUALITY ASSURANCE APPROACH", New Trends in Information and Service Science, 2009. NISS '09. International Conference on

6 Author CVs

Sajid Ibrahim Hashmi

Sajid Ibrahim Hashmi is a doctoral researcher at Lero – the Irish Software Engineering Research Centre, supervised by Dr. Ita Richardson. He holds a M.S in Software Engineering from Korea Advanced Institute of Science and Technology (KAIST). His research interests include software engineering for service-based systems with particular focus on quality assurance and management.

Stephen Lane

Stephen Lane is a doctoral researcher at Lero – the Irish Software Engineering Research Centre, supervised by Dr. Ita Richardson. He holds a M.Tech. in Computer Integrated Manufacturing from the University of Limerick. His research focuses on software engineering processes for service-based systems with a particular interest in adaptability and maintenance.

Dimka Karastoyanova

Dr. Dimka Karastoyanova is an associate professor at the Institute of Architecture of Application Systems at the University of Stuttgart. Her research includes applying the workflow technology in scientific simulation, Web Services (WS) and the related standards and technologies, Business Process Modelling, techniques for enabling adaptability of business processes, in particular BPEL processes, extensibility and parametrization constructs promoting reuse, flexibility and composability of processes, methodologies for creating and executing reusable WS-flows with run time flexibility, toolkit for supporting the automated development of WS-flow definitions, their execution and analysis, transaction processing and workflow management and thier interrelationships, semantic description of WSs and WS-flows and Middleware for Semantic Web Services.

Ita Richardson

Dr. Ita Richardson is the Research Area Leader of Practice, Process and Methods within Lero – the Irish Software Engineering Research Centre and a Senior Lecturer with the Department of Computer Science and Information Systems at the University of Limerick, Ireland. Her research focuses on software quality and software process improvement with an interest in specific environments such as global software engineering, health care and medical devices, small to medium sized enterprises and software services. Much of her research is carried out in industry and in Government-funded services, particularly in Education and Health.