Report of the 2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA 2012)

Grace A. Lewis
Carnegie Mellon Software Engineering Institute
Pittsburgh, PA USA
glewis@sei.cmu.edu

Anca Ionita
University Politehnica of Bucharest
Bucharest, Romania
Anca.Ionita@aii.pub.ro

Marin Litoiu
York University
Toronto, ON Canada
mlitoiu@yorku.ca

DOI: 10.1145/2439976.2439998
http://doi.acm.org/10.1145/2439976.2439998

ABSTRACT

The 2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA 2012) was held as a co-located event of the International Conference on Software Maintenance, ICSM 2012, on September 24, 2012. MESOCA 2012 brought together software engineering researchers from academia and industry, as well as practitioners, to share results and open issues in the area of maintenance and evolution of service-oriented and cloud-based systems. The many discussions throughout the day resulted in the identification of promising areas of research to address existing gaps and problems.

Categories and Subject Descriptors

D.2.11 [**Software Architectures**]: *Patterns – Service-Oriented Architecture*, H.3.4 [**Systems and Software**]: *Distributed Systems – Service-Oriented Systems*

General Terms

Maintenance, Evolution, Design

Keywords

Software maintenance, software evolution, services, SOA, service-oriented architecture, service-oriented systems, cloud computing, cloud-based systems

1. INTRODUCTION

There are many successful case studies of SOA (Service-Oriented Architecture) adoption, mainly in commercial enterprises. Part of what Gartner terms the "period of enlightenment" related to hype cycles for emerging technology is the move from SOA as simply a set of technologies to service-orientation as a mindset for architecting, implementing and deploying services that add value to an organization. Regardless of this positive perception change, there are still two concerns from a maintenance and evolution perspective: (1) deployed service-oriented systems will have to be maintained and evolved and (2) legacy systems will continue to use service-orientation to make their legacy functionality available to other systems and applications.

Cloud Computing is emerging as a model for system development and deployment, in which systems use resources from — or are hosted, run and managed — in large server farms and data centers, and provided as a service. The lack of control over these external resources creates additional challenges for systems migrating to these environments.

The main goal of MESOCA 2012 was to continue to be a focal point and a forum for researchers and practitioners to share results and open issues in the area of maintenance and evolution of service-oriented and cloud-based systems. The workshop brought together five paper presenters, two keynote speakers, one invited speaker and three organizers from nine countries, in addition to other workshop attendees.

A tag cloud built from the different abstracts using TagCrowd[1] and shown in Figure 1 shows some interesting observations in addition to an insight into the depth of breadth of topics that would be addressed during the workshop:

- Multiple tags related to "foundation elements", such as architecture, legacy, environment, model, framework, process, infrastructures, and requirements, show that there is still work to be done in establishing baselines for migration and evolution.
- Multiple tags related to "operational aspects" such as business, service, performance, security, SaaS, resources, support, and runtime shows a concern for system qualities before, during, and after any migration and evolution effort.
- Not surprisingly, and consistent with discussions at previous instances of MESOCA, the technology tag is very small, showing that process, operational, and cultural aspects are more challenging than the technologies used for migration and evolution.

Figure 1. Tag Cloud of Abstracts

2. STRUCTURE OF THE WORKSHOP

The workshop started with a keynote by Carl Worms, an enterprise architect at Credit Suisse Private Banking IT, with a focus on strategy and architecture of software engineering processes. The title of the keynote was "SOA and Cloud — Experiences from a Large Enterprise." There were two paper sessions, one focusing on migration to SOA and cloud environments and one on challenges of cloud environments. Marin Litoiu, a professor from York University, gave an invited presentation of his work on smart applications on cloud infrastructures. To close the workshop, Leire Orue-Echevarría Arrieta, the lead for TECNALIA's "Migration to Cloud (SaaS)" product, presented a keynote titled "From Software as a Good to Software as a Service:

[1] http://tagcrowd.com/

Preparing the Evolution of Software Products into the Cloud." Each session was followed by open discussions. The main points of these discussions were captured by the workshop organizers and are presented in this report.

3. SUMMARY OF PRESENTATIONS

The slides for the keynotes and several of the talks that were presented at the workshop are available on the MESOCA 2012 web site[2]. Full papers are available in the workshop proceedings [1].

3.1 Opening Keynote on SOA and Cloud — Experiences from a Large Enterprise: Carl Worms

The opening keynote was given by Carl Worms from Credit Suisse. The goal of the talk was to tell the story of 15 years of experience with SOA and some 5 years with cloud computing in a global enterprise that is one of the technology leaders in its industry.

The talk started by presenting some data about Credit Suisse to get an understanding of the size of its IT department. They have one central IT division with people in 64 different legal entities globally. The company has 50,000 employees and one-third are either internal or external IT, which to him is not surprising given that banks nowadays are about data processing. They maintain 6,000 different applications and have four releases per year, which is why they often call themselves a "maintenance shop."

To manage a department and application portfolio of this size they have created a set of IT architecture disciplines that roughly comprise three horizontal layers that correspond to business, application and technical/platform; four cross-cutting layers that correspond to data, integration, security, and systems management; all surrounded by IT architecture governance and processes.

Credit Suisse has been involved with the different cloud computing service models over the past five years, all in a private cloud deployment model. As part of IaaS (Infrastructure as a Service) they have a Compute Hosting Platform (CHP), and as part of PaaS (Platform as a Service) they have a DB (Data Base) Hosting Platform and an Application Platform (AP). As part of SaaS (Software as a Service) they have several applications and have just started looking at public cloud for applications that do not manage sensitive and/or customer information, such as order management. Banking is delicate because trading algorithms and pricing are current differentiators in this industry.

The main cloud use cases that they see are application development and operations; in particular ordering capacity, deployment, testing (several stages), and infrastructure change management. The main challenge then becomes packaging applications to work on the pre-provisioned platforms.

The approach that they have taken for cloud design can be summarized as:

- Top-down design — standardized features
- Think-out-of-the-box for new environment — avoid legacy lock-in
- Use unchanged vendor products for design exploration and proofs of concept
- Strive for radical simplicity — avoid early optimization of last 5-10%
- Make everything automatable
- Hide actual implementations/products behind abstracted service APIs
- Design for use in all environments (test/secure server/DMZ, branches)

This has led to a series of recommended cloud design principles because in his opinion clouds will force you to layer applications.

- Location independence
- Placement independence
- Cloning
- Dynamic in-place configuration
- Horizontal scalability
- Capacity is allocated in fixed-sized chunks
- Eco-system automation with service APIs
- High degree of automation
- No client login
- Infrastructure service APIs
- Avoid vendor dependencies

Given all the challenges and all they still have to learn about cloud, they expect migration to cloud to take 10 to15 years, similar to what it took them to migrate from mainframes to a client/server environment.

Their SOA experience started in 1997 with CORBA[3] and the Credit Suisse Information Bus. They moved to Global SOA in 2005 along with DiMA (Disentangling the Mainframes) and to Enterprise SOA in 2008.

The main components of their SOA design and implementation and development approach are:

- Decomposition into components — This means to expose a business view that not coupled to database design or existing data structures.
- Credit Suisse eXchange Bus (CSXB) — This includes the Interface Management System (IFMS), service integration mechanism (synchronous, asynchronous and bulk), portal integration for service composition across business domains, and a OneBank BPM Platform (OBPM).
- Central service repository — IFMS is key to manage 1,100 services. The tool was developed in-house and serves as a service catalog, design tool, governance enforcer, lifecycle manager and code generator.
- SOA-related roles — These are represented by the IFMS user community: service designer, service reviewer, service developer, and service portfolio manager.
- SOA governance — Governance is mainly implemented as a three-step quality assurance process: integration architecture team (do we need it?), cross-functional review team (is it well designed?), tool and process support (is it implemented as specified?)

Their reuse experience with service orientation over these years can be summarized as:

- Availability drives use — Currently 80% of transactions implemented via services.
- Reuse is very uneven — Average reuse is 4 (four different applications use a service).

The challenges and next steps for Credit Suisse are to establish PaaS for standard application platforms on top of their IaaS and the globalization of SOA within Credit Suisse, which means standardization of data structures, global support organizations, and the technical migration of 2600 CORBA services to Web services.

The talk concluded with a proposed set of research topics for the community:

- Cloud topics
  - Empirical research on resource clustering that focuses on CPU, memory, and network usage and optimization
  - Efficient redesign of software for "cloudability"
  - Software quality expressed as measurable properties of "cloudability"
  - Testing for and in the cloud
- SOA topics
  - Security, specifically alternatives for passing original initiator credentials through all (session-less) instances and fine-grained access control

[2] http://www.sei.cmu.edu/community/mesoca2012/

[3] http://www.corba.org

- Management of large service networks, including testing of complex component networks and accounting of service usage
- Design and architecture of high volume, low latency implementations, including investigation of particular hardware such as FPGA (field-programmable gate arrays) and simplified protocols

3.2 Session on Migration to SOA and Cloud

A recent survey of cloud adopters showed that (1) for 60% of respondents cloud applications are better than those on premises in terms of availability, total cost of opportunity (TCO), and time to value (TTV), and (2) 64% said they would have most applications in the public cloud in the next three years [2]. However, this same survey states that (1) for more than 75% of respondents, cloud- to-cloud integration and mobility access are the main challenges and (2) only 4% have integrated cloud applications. Because one of the main advantages of SOA adoption is ease of integration between systems, the goal of this session was to explore service orientation as a complement to cloud computing, more specifically the role of processes, the role of policies, and whether SOA is the answer. There were three papers in this session that related to these questions.

- Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud (*Nabor Mendonca, University of Fortaleza (UNIFOR), Brazil*[4]): Cloudstep is the result of joint work between the software process and reengineering groups at the university. The motivation for Cloudstep is the lack of a systematic process to guide project managers and application developers in making informed cloud selection and migration decisions. Cloudstep is a process that helps with the identification and analysis of relevant cloud migration factors. Feasibility/matching between organizations and could environments is based on profile-based characterization of organizations, applications, and cloud providers (more applicable to IaaS, but can be used for PaaS or SaaS). Constraints include financial, organizational, security, communication, performance, availability and suitability. Non-successful outcomes include stop migration, change the application, change the migration scope or change the cloud provider. Successful outcome is a migration strategy that includes migration costs as well as operational and management costs — actual migration is outside of the scope of this process. Nabor also presented a case study at a Brazilian company called Naja RIS that compared two scenarios in four settings (on premise, Amazon EC2 Virginia, Amazon EC2 Sao Paulo, Rackspace). Future work will integrate results of case study, migration to multi-cloud environments and tool support.
- Linking Legacy Services to the Business Process Model (*Stephan Sneed, Metasonic AG, Germany*): This presentation discussed a real-world reverse engineering project for converting legacy COBOL code to a business process model. The goal of the project was to provide a better basis for the maintenance of service-oriented systems and to allow impact analysis to traverse the border between model and code. The Eclipse-based tool suite is based on EMF (Eclipse Modeling Framework). It takes in COBOL code and using XML[5]- and WSDL[6]-based transformations produces a business model expressed in the S-BPM modeling language, including its corresponding natural language description. Future development will focus on extending the tool to support linking of Java and .Net systems.
- Policy Modeling and Compliance Verification in Enterprise Software Systems: A Survey (*Georgios Chatzikonstantinou, National Technical University of Athens, Greece*): Georgios presented the results of a survey and classification of policy modeling and policy compliance verification techniques, along with pros and cons of each. The work focused four policy types: security, business process, regulatory, design. In the resulting taxonomy, policy modeling techniques were grouped into graphical representations (UML profiles, sequence charts, directed graphs, agent-based) and those based on formal languages (logic-based, high level). Policy compliance verification techniques were classified as model checkers, probabilistic model checkers, and theorem provers. The goal of this work is to be able to support more complex or elastic computing patterns that facilitate the provision of on-demand computing resources. An element of this pattern is a service management and service assurance framework in which policies are formally modeled and consequently verified against runtime system behavior models. Future research will focus on combining reverse engineering with monitoring techniques to verify that the system complies with a set of policies, tracing events against compliance constraints to identify deviations from service-level agreements (SLAs), and tracing actual resource usage patterns for automatic system reconfiguration at runtime.

3.3 Session on Challenges of Cloud Environments

Challenges of cloud environments include selecting a cloud provider, configuration and deployment of virtual machines, runtime adaptation strategies, selecting database management systems to deploy in cloud environments, solutions for massive data repositories, performance and security of cloud architectures, and transforming existing applications to work in or with cloud environments. There were two papers in this session that addressed some of these challenges.

- CDOSim: Simulating Cloud Deployment Options for Software Migration Support (*Florian Fittkau, University of Kiel, Germany*): Simulation can be used to help find the best trade-off between high performance and low cost. A cloud deployment option (CDO) is the combination of decisions regarding cloud provider selection, component-to-VM deployments, VM instance configuration and specific adaptation strategies (per layer). CDOSim is the evaluation step of a larger migration process for enterprise software called CloudMIG, available at http://www.cloudmig.org/. CDOSim is built on top of CloudSim, a cloud provider simulation. It uses the MIPIPS Benchmark — million integer instructions per second — to measure computing performance of a VM instance. Simulation is based on a combination of static and dynamic analysis. The output of the tool is cost, response time, SLA violations and rating. As a validation of the work, the MIPIPS benchmark was compared against two other benchmarks and calculated the relative error between simulated and actual values.
- A Three-Dimensional Data Model in HBase for Large Time-Series Dataset Analysis (*Eleni Stroulia, University of Alberta, Canada*): A general problem is how to organize data in HBase[7] as an example of a novel application stack. The expected outcome of this work is a set of design patterns for HBase. HBase is the open source version of Google's BigTable [3]. It is a distributed, 3D table data structure in which time stamp is the third dimension. Each table has one or more "column families" that are stored as one file in the Hadoop file system (HDFS)[8]. This work is relevant for data migration to the cloud, mainly for applications that have time-based data, such as geographic applications. They experimented using two large data sets: Cosmology (astrophysics particle data) and Bixi (bike rentals in Canada). Early results show that there are differences between static and dynamic data, the amount of historical data, and localization (to take advantage of parallelism). The challenge is to design the right key, which should

[4] The presenter of each paper is cited in this report. For a complete list of papers including credit to multiple authors, see [1].
[5] http://www.w3.org/XML
[6] http://www.w3.org/TR/wsdl
[7] http://hbase.apache.org/
[8] http://hadoop.apache.org/

account for the natural period for querying data (e.g., hour, day, month, quarter, year).

3.4 Invited Presentation on Smart Applications on Cloud Infrastructures: Marin Litoiu

Marin Litoiu presented his current work at York University on Smart Applications on Cloud Infrastructures. The focus of this work is partitioning applications for two-tier clouds: Tier 1 (edge) clouds deployed close to end users and Tier 2 cloud (core), a public cloud. Tier 1 can be a private cloud or the edge of a distributed cloud. The rationale is that centralized clouds cannot support all types of applications. In addition to the implementation of two tier clouds (edge and core), the goal of the project is to provide integrated end-to-end elasticity. These clouds can be exploited by smart apps that can sense their environment and adapt at runtime based on current and future load.

Applications are partitioned into what to execute locally (private cloud/edge) and what can be "bursted" to the public cloud (core), using a small set of annotations that are then expanded using dependency graph analysis to exactly determine what code units can be moved. The moment at which bursting occurs is determined by a monitoring service that monitors system load and anticipates future load. There is a working implementation for Java and PHP code.

During the discussion there were many questions about where data resides. In the current implementation, the code marked as public should not depend on data in the database because that remains local if the driver is performance. In their case, their driver is privacy and therefore the data should stay local. A next step could be to find ways to annotate the database as well so that the bursted code can move along with its data — the next challenge would be data synchronization.

3.5 Closing Keynote on From Software as a Good to Software as a Service: Preparing the Evolution of Software Products into the Cloud: Leire Orue-Echevarría Arrieta

The closing keynote was from Leire Orue-Echeverría Arrieta from TECNALIA. The goal of the talk was to share their experience of creating and applying a proposed architecture and process for companies that are moving from being a software-as-a-product based company to a software-as-a-service based company.

The keynote started by stating that the business model changes greatly in this type of evolution and creates new questions and challenges:

- Is the cloud suitable for me?
- How to adapt my applications to the cloud?
- Which functionality/modules should I add to my application to offer it as a service over the cloud?
- And what if I want to change IaaS providers?
- How do I know my IaaS providers is meeting established SLAs

In addition to these general challenges, this evolution towards a SaaS company creates organizational, requirements and architectural challenges:

- Organizational challenges
  - ROI (Return on Investment) and payback
  - It is not everyday's work — new roles, responsibilities, tasks, competencies, processes
  - No expertise within the organization — everything is new
  - Provider lock-in
- Requirements challenges
  - Maintaining requirements from the original legacy system
  - Meeting new non-functional requirements
  - Technological challenges
  - Lack of SOA support
  - Lack of system-of-systems validation support
  - Lack of support for SaaS-compliant requirements
  - Lack of model-driven-engineering support
  - Multiple graphical user interfaces (GUIs)
- Architectural challenges
  - System needs to be adapted to be SaaS-compliant
  - No "one size fits all" when it comes to reuse
  - Unpredictable performance
  - Procedural challenges
  - Different maintenance, deployment and support procedures
  - Demand provisioning procedures
  - Dependency management
  - New withdrawal procedures — e.g. how to break the contract and get your data back

To address some of these challenges they created the Migration to SaaS (M2S) process, which consists of the following stages:

- Pre-migration: Includes tool support for the migration and provider decision, maturity assessment (questionnaire) and feasibility analysis (technical and business, source code analysis and cost-benefit analysis)
- Migration: Includes support for the actual migration in the form of recovery (model extraction), implementation (model transformation, code generation, new code) and V&V (test case identification and execution, compare results with test data for legacy system)
- Provisioning: Includes tool support for provisioning and maintenance, which might require the creation of a certification model to create consumer trust

4. MAIN OUTCOMES AND NEXT STEPS

As in past instances of MESOCA, there was a lot of discussion on the maintenance and evolution of service-oriented and cloud-based systems, this time triggered by the research challenges identified by Carl Worms in the opening keynote.

Based on these discussions, there is general agreement that SOA and cloud are complementary, in the sense that service orientation as a development model is a good match for cloud computing as a deployment model, because it provides the integration, elasticity, flexibility and monitorability that is required of systems deployed in the cloud. Because of this fact, it was also agreed that most of the research that was discussed in the context of service-oriented systems is also applicable to cloud-based systems. Many systems are migrating and will continue to migrate to these environments as a natural form of evolution to new computing paradigms. If only as an indicator of barriers to cloud computing adoption, data privacy and data location were the main concerns of the attendees.

Given the continued support and participation of the software maintenance community, MESOCA 2013 will be co-located with ICSM 2013 in Eindhoven, The Netherlands. The event has evolved from a workshop to a symposium to better match the maturity and contributions that the venue has made in the field. We invite researchers and practitioners to submit publications and attend. More information can be found at http://www.sei.cmu.edu/community/mesoca2013/.

5. REFERENCES

[1] Ionita, A.; Lewis, G.A.; Litoiu, M., "2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)," IEEE Conference Publications, 2012, URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6380663

[2] Narasimhan, B.; Nichols, R. "State of Cloud Applications and Platforms: The Cloud Adopters' View," Computer, vol. 44, no. 3, pp. 24-28, March, 2011.

[3] Chang, F.; Dean, J.; Ghemawat, S.; Hsieh, W.; Wallach, D.; Burrows, M.; Chandra, T.; Fikes, A.; and Gruber, R. Bigtable: a distributed storage system for structured data. Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7 (OSDI '06), Vol. 7. USENIX Association, Berkeley, CA, USA, 15-15. 2006.