SUMMARY

28TH IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE (ICSM 2012)

RIVA DEL GARDA, TRENTO, ITALY
SEPTEMBER 24, 2012

GENERAL CHAIR: GRACE A. LEWIS
PROGRAM CO-CHAIRS: ANCA IONITA AND MARIN LITOIU
Opening Keynote: SOA and Cloud — Experiences from a Large Enterprise
Carl Worms, Credit Suisse AG

- One central IT division with people in 64 different legal entities globally — 50,000 employees where 1/3 company employees are either internal or external IT — not surprised, banks nowadays about data processing — 6,000 applications — 4 releases per year, which is why they call themselves a “maintenance shop”
- IT architecture disciplines: three horizontal layers: business, application and technical/platform; four cross-cutting layers: data, integration, security and systems management — all surrounded by IT architecture governance and processes
- 5 years of Cloud @ Credit Suisse (private cloud)
  - IaaS: Compute Hosting Platform (CHP)
  - PaaS: DB Hosting Platform, Application Platform (AP)
  - SaaS: Applications — 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
- Main Cloud Use Cases is application development and operations — ordering capacity, deployment, testing (several stages), infrastructure change management — main challenge is packaging applications to work on the pre-provisioned platforms
- Cloud design
  - Top-down design approach — 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)
- Cloud design principles — 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
- Expect migration to cloud to take 10-15 years, similar to what it took them to migrate from mainframes to client/server
15 years of SOA @ Credit Suisse
- Part of the integration architecture
- Started in 1997 with CORBA and the Credit Suisse Information Bus — moved to Global SOA in 2005 along with DiMA (Disentangling the Mainframes) — Enterprise SOA in 2008
- Main components
  - Decomposition into components — expose a business view — not coupled to database design or existing data structures
  - Credit Suisse eXchange Bus (CSXB) — includes an Interface Management System (IFMS), service integration mechanism (sync, async 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 — developed in-house — serves as a service catalog, design tool, governance enforcer, lifecycle manager and code generator
  - SOA-related roles — the IFMS user community — service designer, service reviewer, service developer, and service portfolio manager
  - SOA governance — quality assurance process — three step process: integration architecture team (do we need it?), cross-functional review team (well designed?), tool and process support (implemented as specified?)

Experiences
- Availability drives use — 80% of transactions implemented via services
- Reuse is very uneven — average is 4 (4 different applications use a service)

Challenges and next steps
- Establish PaaS for standard application platforms on top of their IaaS
- Globalization of SOA within Credit Suisse — standardization of data structures, global support organizations, technical migration of 2600 CORBA services to Web services

Research topics (see presentation)
A recent survey of cloud adopters shows the advantage of cloud applications over on-premise applications and the trend for migrating to the cloud. But, there are challenges, mainly related to application-to-application integration and mobility access.

Discussion points:
- What is the role of processes?
- Is SOA the answer for integration?
- What is the role of policies?
Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud
Nabor Mendonça (University of Fortaleza (UNIFOR), Brazil)

- Joint work between software process and reengineering groups at the university
- Driven by lack of a systematic process to guide project managers and application developers in making informed cloud selection and migration decisions
- Cloudstep helps with the identification and analysis of relevant cloud migration factors
- Feasibility/matching 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
- Case study at a Brazilian company called Naja RIS compared two scenarios in 4 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
Reverse engineering project for converting legacy COBOL code to a business process model in S-BPM, via XML and WSDL

Additional step is translating the S-BPM model to natural language

S-BPM is a Eclipse-based tool suite — uses EMF (Eclipse Modeling Framework) — takes in a WSDL file and produces corresponding natural language
### Policy Modeling and Compliance Verification in Enterprise Software Systems: A Survey

Georgios Chatzikonstantinou (National Technical University of Athens, Greece)

<table>
<thead>
<tr>
<th>Points</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey and classification of policy modeling and policy compliance verification techniques, along with pros and cons of each</td>
<td>Focus on four policy types: security, business process, regulatory, design</td>
</tr>
<tr>
<td>Policy classification</td>
<td><strong>Graphical:</strong> UML profiles, sequence charts, directed graphs, agent-based&lt;br&gt;<strong>Formal languages:</strong> logic-based, high level</td>
</tr>
<tr>
<td>Policy compliance verification technique classification</td>
<td><strong>Model checkers</strong>&lt;br&gt;<strong>Probabilistic model checkers</strong>&lt;br&gt;<strong>Theorem provers</strong></td>
</tr>
<tr>
<td>Future research</td>
<td>Combine reverse engineering with monitoring techniques to verify that the system complies with a set of policies&lt;br&gt;Tracing events against compliance constraints to identify deviations from SLAs&lt;br&gt;Tracing actual resource usage patterns for automatic system reconfiguration at runtime</td>
</tr>
</tbody>
</table>
Challenges of Cloud Environments

- Selecting a cloud provider
- Virtual machines — configuration and deployment
- Runtime adaptation strategies
- Selecting a database management system
- Solutions for massive data repositories
- Performance and security with Cloud architectures
- Transforming existing applications
Simulation too to help find the best trade-off between high performance and low cost

Built on top of CloudSim, a cloud provider simulation and part of a larger approach called CloudMIG (CDOSim is the evaluation step) — available at http://www.cloudmig.org/

Context is enterprise software

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)

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

Output is cost, response time, SLA violations and rating

Evaluated the MIPIPS benchmark against two other benchmarks — 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)

- General problem is how to organize data in HBase as an example of a novel application stack — expected outcome of this work is a set of design patterns for HBase
- HBase is the open source version of BigTable — it is a distributed, 3D table data structure — 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)
- Work is relevant for data migration to the cloud, mainly for applications that have time-based data, such as geographic applications
- 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)
- Challenge is to design the right key — should account for what the natural period is for querying data (e.g. what is the most recent data for X? Day? Month? Year?)
Focus is partitioning applications for two-tier clouds — rationale is that centralized clouds cannot support all types of applications.

Tier 1 (edge) clouds deployed on routers are connected to the Tier 2 cloud (core).

Project goals:
- Two tier clouds: edge and core
- Integrated end-to-end elasticity
- 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.

Currently implemented for Java code.

Lots of questions about where the data resides — the code you mark as public should not depend on data in the database because that remains local if the driver is performance (their driver is privacy) — next step could be to find ways to annotate the database as well so that the bursted code can move along with its data — of course the next challenge would be data synchronization.
- **Problem statement:** going from software-as-a-product based company to a software-as-a-service based company
  - The business model changes greatly!

- **General 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

- **Organizational challenges**
  - ROI and payback
  - Not every day'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 SoS validation support
  - Lack of support for SaaS-compliant requirements
  - Lack of MD(r)E support
  - Multiple GUIs
Closing Keynote: From Software as a Good to Software as a Service: Preparing the Evolution of Software Products into the Cloud

Continuation ...

- **Architectural challenges**
  - System needs to be adjusted to be SaaS-compliant
  - No “one size fits all” when it comes to reuse
  - Unpredictable performance

- **Procedure 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

- **M2S — Migration to SaaS**
  - Pre-migration: Tool support for migration and provider decision — maturity assessment (questionnaire) and feasibility analysis (technical and business, source code analysis and cost-benefit analysis)
  - Migration: Tool support for the migration — 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: Tool support for provisioning and maintenance — might require creation of a certification model to create consumer trust
• Introduction and summary slides will be on the web site: http://www.sei.cmu.edu/community/mesoca2012/
• Presenters, please send me your presentations in PDF format so I can put them on the web site
• Get ready for MESOCA 2013!
Co-located with ICSM 2013 in Eindhoven, The Netherlands
Tentative date: September 23, 2013
General Chair: Anca Ionita
We want to open it to the community
- Increase size and diversity of program committee
- Increase size of the event
- Open the chair roles to members from different organizations
- Please send us your ideas

SEE YOU NEXT YEAR! TELL YOUR FRIENDS!