Yudistira Asnar - University of Trento, Italy

Yudistira Asnar is a research fellow at University of Trento. He received BEng, in 2002 from ITB, Indonesia and PhD on security risks on secure and dependable systems at University of Trento, Italy in 2009. He visited the Open University, UK in 2007. His research interests lie in the area of requirement engineering, agent systems, security-dependability risk management, and information assurance. The main focus of his research is on the modeling and analyzing governance, risk & compliance of IT services.

Hoon Wei Lim - SAP, France

Hoon Wei Lim is a SAP Researcher based on Sophia Anti Polis, France. He obtained a Ph.D degree in Information Security from Royal Holloway, University of London in 2006. His doctoral research focused on various key management and security architectural issues for grid computing systems. Upon completion of his PhD, Dr. Lim stayed on at Royal Holloway and worked as a post-doctoral research assistant in an e-Science project funded by the UK EPSRC. Lim is now a researcher with SAP Labs France, working on EU-funded projects related to security, privacy and compliance.

Fabio Massacci – University of Trento, Italy

Fabio Massacci received a M.Eng. in 1993 and Ph.D. in Computer Science and Engineering at University of Rome La Sapienza in 1998. He joined University of Siena as Assistant Professor in 1999, and was visiting researcher at IRIT Toulouse in 2000, and joined Trento in 2001 where is now full professor. His research interests are in security requirements engineering, formal methods and computer security. He was deputy rector for ICT procurements with a multimillion-euro budget and currently is currently scientific coordinator of several industry lead R&D European projects on security and compliance.

Claire Worledge - Deloitte, France

Claire Worledge has worked for Deloitte for the past 8 years and is currently manager in the Paris office. She is a specialist in Computer Assisted Audit Techniques (CAAT) and works on projects ranging from financial audit support and continuous controls monitoring to fraud detection. Claire also provides support to Project Management Office teams during implementations of Enterprise Resource Planning systems. Before working for Deloitte, Claire attained a Masters in Information Technology from University College London and a Masters in Information Security from Royal Holloway London.

Realizing Trustworthy Business Services by A New GRC Approach

Introduction

The trustworthiness of business services is widely recognised as a critical factor for the success of an organization. Businesses are increasing in complexity and unpredictability, while demand for accountability, as well as regulatory compliance is becoming mandatory. Yet, reports[i] indicate that the level of fraud within an organization is far from decreasing.. Thus, a structured approach to

Governance, Risk and Compliance (GRC) has become a high priority goal for many organizations [2]: "Governance" is the policies, laws, culture, and institutions that define how an organization is managed; "Risk Management" concerns the coordinated activities that direct and control an organization's risks; "Compliance" is the act of adhering to regulations as well as corporate policies and procedures.

GRC solutions[ii] enable organizations to address various business challenges related to risk management and regulatory compliance. For example, GRC solutions provide end-to-end control management, deployment of controls through risk-based approaches and automatic monitoring of controls across different entities and applications. Furthermore, GRC solutions enable standardization of methodologies, vocabulary and measurements across an organization, therefore facilitating the detection of risks, the prioritization of corrective actions and so the enforcement of compliance.

Challenges of Services

Despite a better understanding of the GRC challenges in monolithic systems, new challenges emerge from the implementation of IT systems using Service-Oriented Architecture (SOA) technologies. SOA improves the flexibility and scalability of business solutions [1]. By means of SOA, business processes may be executed and connected through web services that can be modified or adapted quickly in response to changing business market requirements. Vendors of Enterprise Application Integration (EAI) and Business Process Management (BPM) products integrate their proprietary technology with standardized, service-based interfaces and processes [3]. SOA platforms support the integration of services and components across organizational domains, and enable their reuse in different business settings via a simple composition. Thus, SOA provides an IT infrastructure that supports dynamic outsourcing and the integration of business ecosystems. SOA also enables the adaptation of business processes and applications as well as their response to changing requirements and contexts.

Despite this market trend, existing GRC solutions do not yet take into consideration the additional risks associated with SOA-based business environments. For example, how can a finance manager

obtain assurance that the services supporting the finance business processes are trustworthy? How can s/he monitor the behaviour of services underlying a business process?

The adaptability and flexibility of SOA introduces additional challenges for traditional GRC approaches [4]:

- Abstraction: a crucial feature of SOA is that services can be accessed through an abstract interface. The abstraction levels of control objectives and service interfaces is not necessarily the same. An explicit mapping is needed when control objectives are imposed on a service.
- Dynamics and Flexibility: SOA supports the continuous change of business relations (i.e., services provided and consumed) and business processes (the orchestration of the services). Each change potentially violates control objectives or influences the effectiveness of controls. Therefore, control monitoring and evaluation should be a continuous process.
- Distributed control: a fundamental principle of SOA is the possibility to discover and integrate services of different providers at runtime. From the consumer point of view this means that controls may not be directly imposed on alien services. It is therefore necessary to be able to determine which alien services really need to be controlled and how the controls impact the achievement of control objectives.
- Evolving perimeter: several business strategies (e.g., outsourcing, strategic alliance) require an organization to give other organizations (i.e., from service providers in an outsourcing scenario to competitors in a strategic alliance) access to their IT systems. This situation makes some 'classical' security controls (e.g., firewall) ineffective. It is therefore necessary to be able to monitor and control services provisioned by subsidiaries and third parties.

Traditionally, GRC approaches do not offer the level of flexibility, scalability and automation needed for realising trustworthy services. Fortunately, we can use the SOA paradigm itself to facilitate the implementation and monitoring of controls for trustworthy business services.

In the remainder of this article, we describe the MASTER methodology[iii] used to implement GRC on service-oriented business environments. The MASTER methodology is accompanied by an IT

architecture and a set of tools that support: (i) monitoring of events triggered by business services, (ii) analysis and assessment of business service behaviour with respect to control objectives, and (iii) automation of control enforcement.

The MASTER Approach

In general, there are two paradigms for enforcing compliance in the business: (i) *compliance by design* where a business process is designed by considering compliance requirements in addition to business objectives; and (ii) *compliance by control* where a control, a means to comply, is introduced later as a wrapper protecting a business process. Both paradigms have their trade-offs and the discussion about which one is better than the other falls outside the scope of this article. MASTER adopts the latter paradigm since in a SOA environment the design of a system changes over time following the needs of stakeholders and compliance requirements might arrive out of sync with the design & deployment schedule. *Compliance by control* allows each business process owner to employ only necessary controls for the underlying services without major adjustments to the business process itself.

Essentially, the MASTER methodology is founded on three basics concepts: *Risks* that endanger the business operationally or legally, *Controls* to mitigate unacceptable risks, and *Indicators* to monitor the performance and effectiveness of controls. These three basics concepts can be used to improve existing GRC implementations following the Plan-Do-Check-Act of Demming's cycle [5]. Each step of the methodology is detailed in Figure 1.

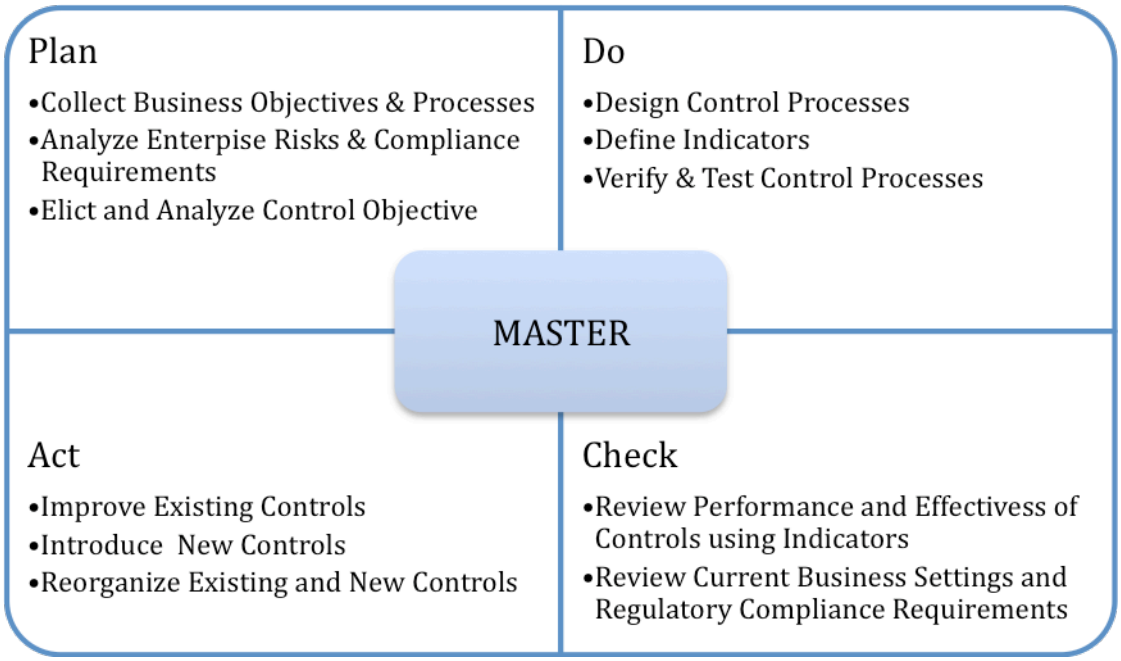


Figure 1 - The PDCA Cycle as applied to the MASTER Methodology

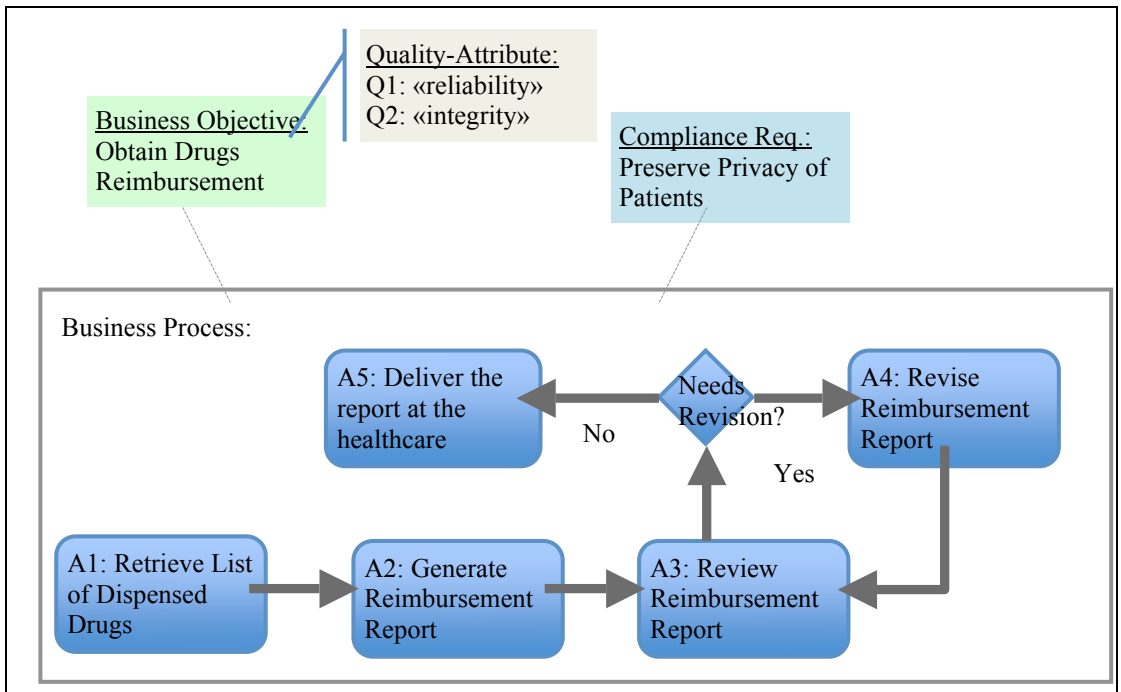
MASTER defines a *control objective* according to the quality-attributes of the business process that is being protected. The technical implementation of a control objective is referred to as a *control process*. In a nutshell, business processes can be seen as the day-to-day workings of the organization, while control objectives and processes help the organization to achieve its business goals (e.g. ensure business processes stay on track). The separation between control processes and business processes is useful as different actors own and are held accountable for these processes. In case of changes to compliance requirements, controls can be modified independently without touching the target business process.

Organizations face the challenge of defining control objectives and control processes that mitigate all of the risks associated with an SOA environment. A good set of control objectives must be CAP: Complete, Accurate and Precise.

- *Complete* – Control objectives must ensure that all critical risks are addressed;
- *Accurate* – Free from errors so that their achievement meets the relevant business goals and mitigates the related risks;

- *Precise* – They must be clearly specified, enabling unambiguous interpretation of the level of compliance or failure of a business process with regards to the control objective.

These 3 qualities are complementary: a control objective might be complete, but not accurate – e.g., it covers all relevant business needs, but, say, wrong security assumptions might let a level of unacceptable risk. The analysis might be accurate (and determine the right effect in terms of impacts and likelihood of harmful events), but the description of the control is not precise enough to allow for the correct implementation or the automation of the solutions. The MASTER methodology ensures that control objectives are CAP through an in-depth and parallel review of pre-determined risks. In Figure 2, we illustrate how control objectives are derived using an example based on a drug reimbursement business process at a hospital.[iv]

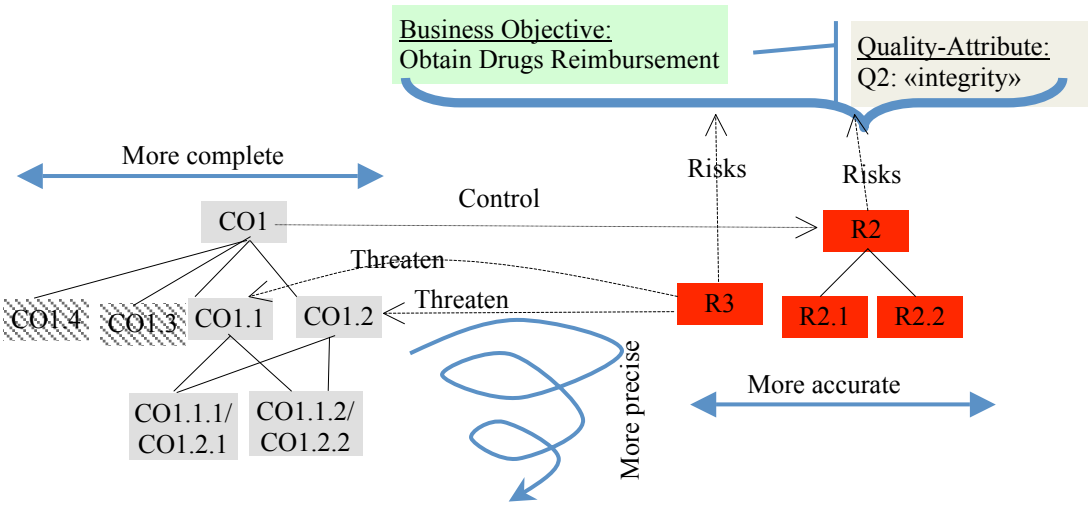


Risk:
R1: Data of dispensed drugs is incomplete → Risk of Q1
R2: New «fake» drugs are added in the reimbursement report → Risk of Q2
R3: Collusion among actors to add new «fake» drugs in the report → Risk of Q2
R4: The hospital and healthcare staff might deduce and expose an individual's disease → Risk of Compliance Req.

Control Objective:
CO1: Ensure all data is complete and correct → preserve Q1 and Q2 from R1, R2, R3
CO2: Hide personal information from the report → fulfil the compliance req. risk (R4)

Figure 2 - Control Objective Analysis

The control objectives CO1 and CO2 specified in Figure 2 might be clear and easy to understand by the stakeholders. However, these control objectives are still not precise enough to be machine implementable and monitored in terms of their effectiveness and performance. Hence, further refinement is required. MASTER adopts a parallel refinement and review model of control objectives and risks, as shown in Figure 3. Each refinement and review iteration of the models leads to an increase in precision, while the broadening of controls increase completeness. More detailed risk analysis improves accuracy of risk estimates and the corresponding mitigation effects.



Risk:
R2: New «fake» drugs are added in the reimbursement report → risk to Q2 of business obj.
R2.1: New drugs are added in the generated report → controlled by CO1.1
R2.2: New drugs are added while revising the report → controlled by CO1.2
R3: Collusion among actors to add new «fake» drugs in the report

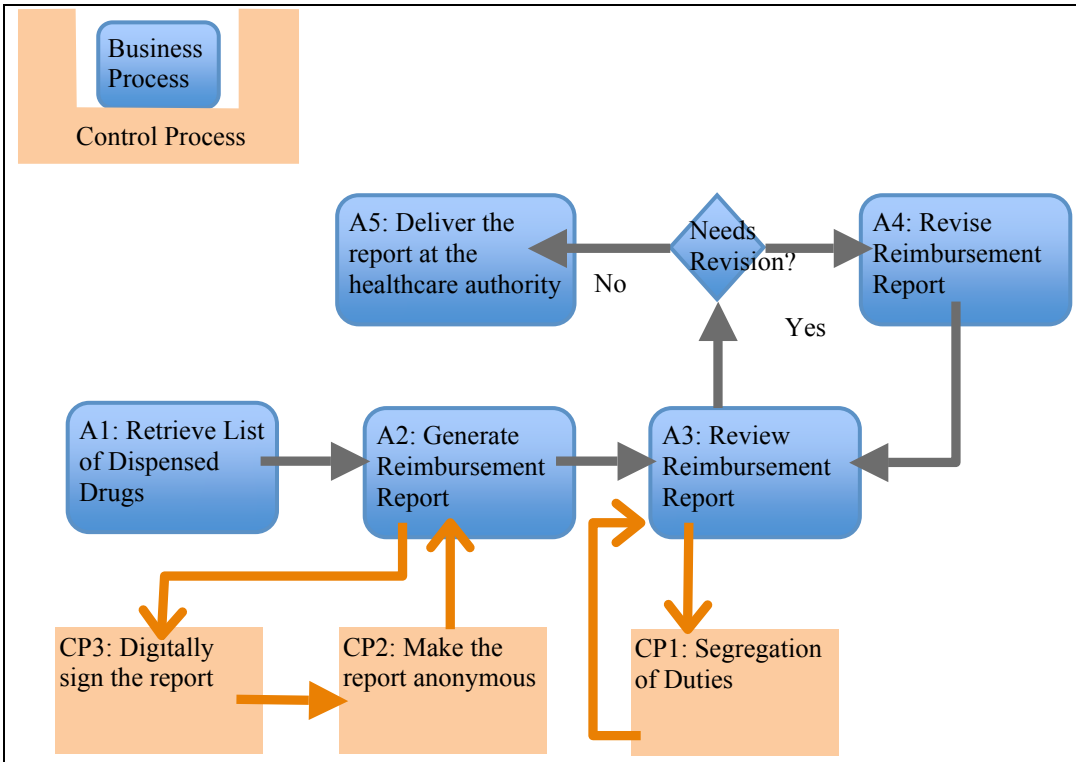
Control Objective:
CO1: Ensure all data is complete and correct
CO1.1: Ensure A2 and A3 are performed by different actors
CO1.1.1: Assign A3 to an actor other than the performer of A2
CO1.1.2: Enforce blind review at A3
CO1.2: Ensure A4 and A3 are performed by different actors
CO1.2.1: Assign A3 to an actor other than the performer of A4
CO1.2.2: Enforce blind review at A3

CO1.3: Digitally sign the report
CO1.4: Review the audit trail by external auditors
Note:
There could be the case a risk (R3) also threatens the effectiveness of control objective (CO1.1 and CO1.2)

Figure 3 - Control Objective Refinement

A control process is then defined as a realization of a control objective (the leaf nodes of Figure 3) and is implemented as a service in a SOA environment as illustrated in Figure 4. In other words, a control

can be seen as a wrapper to the business components (as depicted in Figure 4) to preserve their quality attributes. Once these controls in place, the challenge remains as to how they can be assessed and monitored in real-time.



	Control Processes
CO1.1.1	CP1: Segregation of Duties Prevent access to A3 if the user is the same as that for A2
CO1.1.2	CP2: Make the report anonymous
CO1.3	CP3: Digitally sign the report The report is digitally signed as soon as it is generated by the IT system
CO1.1.4	...out of scope...

*To implement CO1.2.1 and CO1.2.2 one can use the same control processes with different policies. For example, CP1 prevents access to A3 if the user is the same as that for A4.

Figure 4 - Interwoven Control Process and Business Process

For each control objective and processes, analysts need to identify indicators that measure their correctness and effectiveness. For these purposes we introduce two indicators Key Assurance Indicators (KAI) and Key Security Indicators (KSI).

- KAI indicates the effectiveness a control objective in assuring the compliance of business process – E.g., To measure the assurance of *Ensure all data is complete and correct*

(CO1.1.1), we introduce $KAI_{CO1.1.1}$ that measures how many times A2 and A3 are preformed by the same actor.

- KSI concerns the correctness of control process in protecting the business process – E.g., The control of *Segregation of Duty (CP1)* behaves correctly when it rejects the access of A3 if it is done by the same performer as A2. To measure the correctness of CP1, KSI_{CP1} is introduced by measuring how many times CP1 rejects the access of A3 done by A2's performer.

Typically, KAIs are the focus of the business analysts as business analysts are more concerned with the level of compliance rather than how the control is implemented. KSIs, on the other hand, are of interest to risk/security analysts as they measure how well controls are implemented

Both KAI and KSI indicators are critical for monitoring, evaluating, and improving the GRC implementation. The indicators are computed independently in order to distinguish between the case in which the KAI of a control objective is "low" whereas the KSIs associated control processes are "high". In the former case, analysts might conclude that there are some risks that haven't been mitigated. In the latter, it might be that the compliance of a business process is achieved through external factors (from luck to organizational procedures) rather than the deployed controls.

Implementation Guidance

To implement control processes and indicators in an SOA environment, one needs to specify which service events are need to be controlled and monitored. A set of business services is implemented to support the execution of a business process, and likewise for control processes and services. In the above example(Figure 4), A2 (Generate Reimbursement Report) is realized by an application using a web service, namely `GeneratorService`, while A3 (Review Reimbursement Report) uses `ReviewService`. These web services are used to support the overall business process. In addition to the business services, other services, called control services, are implemented to control and monitor the business services, such as: `DigitalSignServices` for CP3, `AnnonymizerService` for CP2,

and `SoDService` for CP1. Essentially, there are two ways a control service works: 1) filter in/out a request to a business service, and 2) verify the output of a business service.

To integrate control services with the business services, it is necessary for these services to be connected through a messaging service (e.g., Java Messaging Service/JMS[v], Enterprise Service Bus (ESB)[vi]). The ESB has a capability to detect when a message (e.g., request, response, notification) arrives, and to perform some actions (e.g., block, delete, delay, release modify, forward). The basic principles for interweaving control and business services are:

- If a control service is executed before the business service is invoked - (i.e., filter in/out)

  The ESB will *block* the request message to the business service and forward the request to the control service. The control service will notify the messaging service whether to *drop* the blocked request if it is considered to be an inappropriate request, or to release it;

- If a control service is executed after the business service invoked – (i.e., verify)

  The ESB will *block* the result of the business service invocation before dispatching it to the subsequent service in the business process, and *release* it after performing some operations (e.g., *modify/add/remove* some data items, attach signature) or even *drop* it if it violates some policy (e.g., not sending confidential data)

Besides implementing control processes, designers need to define the events (e.g., a service start/finish/suspend or messages exchanged among services) that will compute the KAI and KSI. To process these events we can use business activity monitoring (BAM)[vii] since it allows us to analyze real-time events from the business transaction, and furthermore to compute KAIs/KSIs following the mathematical formula defined by the designers.

To implement control processes in Figure 4, we specify a set of policy governing the actions of ESB and BAM.

ESB:

- **Block** every result from `GeneratorService` and **forward** to `DigitalSignServices` to be digitally signed. `AnnonymizerService` emits a *release event* to the ESB after it removes the identity of user generator
- **Release** the results of `GeneratorService` after receiving the *release event* from `AnnonymizerService`
- **Remove** the results of `GeneratorService` when there is no *release event* from `AnnonymizerService` after 4 hour
- **Block** each request to `ReviewService` and forward to `SoDService`. It emits a *release event* if the requester is different from the `GeneratorService's` requester, and emits a *delete event* if otherwise
- **Release** the request to `ReviewService` after receiving the *release event* from `SoDService`
- **Block** the request to `ReviewService` when the *delete event* is received from `SoDService`

BAM: (for CO1.1: Ensure A2 and A3 are performed by different actors).

- KAI –How many times the same actor has performed A2 and A3?
  Count how many times when the requester field of `ReviewService` request, that has been released by the ESB, is the same as the requester field of `GeneratorService` request
- KSI – The percentage of times CP1 rejects access requests to A3 when it the request comes from the A2 performer

$$\frac{N_{delete}}{N_{same-req}}$$

N_{delete} = How many times the `SoDService` emits a *delete event*

$N_{same-req}$ = How many times the requester field of `GeneratorService` request is the same with the one of `ReviewService` request

Final Remarks

We have provided an executive summary of the MASTER methodology and its IT architecture. This methodology, and the related and a set of tools[viii], promotes a GRC approach to implement controls at the service/business process level. This approach is aligned with the abstract interface of SOA, and it improves the flexibility of control process improvement without affecting the business process. A critical aspect of SOA is the support for integration and interoperability of legacy system and applications developed by various vendors. The MASTER methodology allows us to control the execution flow of our business processes that fully exploit these critical features of SOA.

Control processes can therefore be implemented in a distributed environment and assurance is not limited to processes occurring within a single organization boundary.

Acknowledgements

This work was supported by funds from the European Commission (contract N° 216917 for the FP7-ICT-2007-1 project MASTER). We thank Andrea Micheletti and Daniela Marino from San Raffaelle Hospital.

References

[1] Dale Vecchio, “Leverage Your Mainframe Application with SOA”, Gartner Research, October 2005.

[2] M. Rasmussen, “Trends 2007: Governance, Risk and Compliance: Organizations are Motivated to Formalize a Federated GRC Process”, Forrester Research, April 2007.

[3] R. Seeley, “Forrester sees convergence of SOA and BPM”, SearchWebServices.com, 09 Jan 07, available at http://searchsoa.techtarget.com/news/article/0,289142,sid26_gci1238154,00.html

[4] V. Lotz, E. Pigout, P.M. Fischer, D. Kossmann, F. Massacci, and A. Pretschner, “Towards Systematic Achievement of Compliance in Service-Oriented Architectures: The MASTER Approach”, Wirtschaftsinformatik, 50(5): 383-391, 2008.

[5] W.E. Deming, “Out of the Crisis”, MIT Press, 2000.

---

i http://fightfraudamerica.com/

ii https://www.sdn.sap.com/irj/bpx/grc

iii http://www.master-fp7.eu/index.php?option=com_docman&task=doc_details&gid=16&Itemid=60

iv The case study have been kindly provided by Hospital San Raffaele Foundation.. Its complete description is available at http://www.master-fp7.eu/index.php?option=com_docman&task=doc_details&gid=53&Itemid=60

v http://java.sun.com/products/jms/

vi http://www.infoq.com/presentations/Enterprise-Service-Bus

vii http://www.ebizq.net/topics/bam/features/6596.html