Yudistira Asnar, Ph.D., is a research fellow at University of Trento (Italy). His research interests lie in the areas of requirement engineering, agent systems, security-dependability risk management and information assurance. The main focus of his research is on modeling and analyzing governance, risk and compliance of IT services.

Hoon Wei Lim, Ph.D., is a researcher at SAP based in Sophia Antipolis, France, working on EU-funded projects related to security, privacy and compliance. His doctoral research focused on various key management and security architectural issues for grid computing systems.

Fabio Massacci, Ph.D., is a professor at the University of Trento (Italy). He was deputy director for ICT procurements with a multimillion-euro budget and is currently scientific coordinator of several industry-leading research and development projects in Europe on security and compliance.

Claire Worledge is a manager for Deloitte, France. She is a specialist in computer-assisted audit techniques (CAATs) and works on projects ranging from financial audit support and continuous controls monitoring to fraud detection. Worledge also provides support to project management office teams during implementations of enterprise resource planning systems.

Realizing Trustworthy Business Services Through a New GRC Approach

The trustworthiness of business services is widely recognized as a critical factor for the success of an organization. Businesses are increasing in complexity and unpredictability, while demand for accountability and regulatory compliance is becoming mandatory. Yet, reports[1] 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]

GRC solutions[3] enable organizations to address various business challenges related to risk management and regulatory compliance. Furthermore, GRC solutions enable standardization of methodologies, vocabulary and measurements across an organization, facilitating the detection of risks, prioritization of corrective actions, and 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.[4] Vendors of enterprise application integration (EAI) and business process management (BPM) products integrate their proprietary technology with standardized, service-based interfaces and processes.[5]

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 finance managers obtain assurance that the services supporting the finance business processes are trustworthy? How can they monitor the behavior of services underlying a business process?

The adaptability and flexibility of SOA introduces additional challenges for traditional GRC approaches, including:[6]

- **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 are 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. Therefore, it is 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 (e.g., 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., firewalls) ineffective. Therefore, it is 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 realizing trustworthy services. Fortunately, the SOA paradigm can be used to facilitate the implementation and monitoring of controls for trustworthy business services.

The remainder of this article describes the MASTER methodology[7] 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:

- Monitoring of events triggered by business services
- Analysis and assessment of business service behavior with respect to control objectives
- Automation of control enforcement

THE MASTER APPROACH

In general, there are two paradigms for enforcing compliance in the business:

- **Compliance by design**, where a business process is designed by considering compliance requirements in addition to business objectives
- **Compliance by control**, where a control 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 because in an SOA environment the design of a system changes over time. 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.

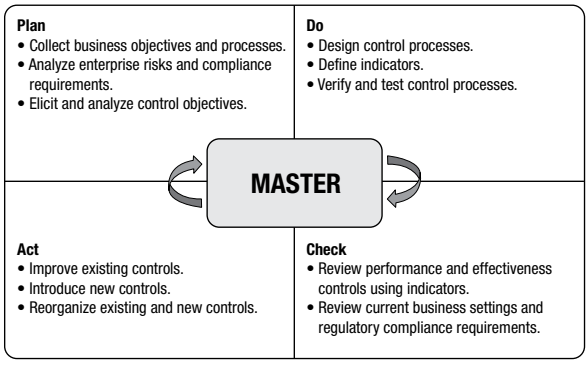
Essentially, the MASTER methodology is founded on three basic concepts:

- **Risks** that endanger the business operationally or legally
- **Controls** to mitigate unacceptable risks
- **Indicators** to monitor the performance and effectiveness of controls

These three basic concepts can be used to improve existing GRC implementations following the Deming Plan-Do-Check-Act (PDCA) cycle.[8] Each step of the methodology is detailed in **figure 1**.

MASTER defines a control objective according to the quality attributes of the business process that are 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 that business processes stay on track). The separation between control processes and business processes is useful as different actors own and

Figure 1—PDCA Cycle Applied to MASTER Methodology



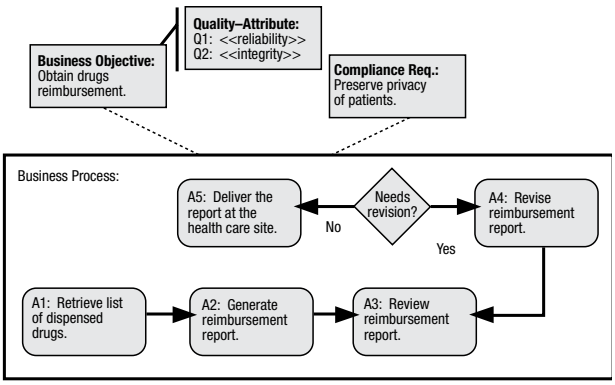
are held accountable for these processes. In case of changes to compliance requirements, controls can be modified independently without touching the target business process.

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 complete, accurate and precise (CAP).

These three qualities are not mutually exclusive; that is, a control objective might be complete, but not accurate. For example, it covers all relevant business needs, but wrong security assumptions might create 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, through an in-depth and parallel review of predetermined risks, that control objectives are CAP. **Figure 2** illustrates how control objectives are derived using an example based on a drug reimbursement business process at a hospital.[9]

The CO1 and CO2 control objectives 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 implemented 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 increases completeness. More detailed risk analysis improves accuracy of risk estimates and the corresponding mitigation effects.

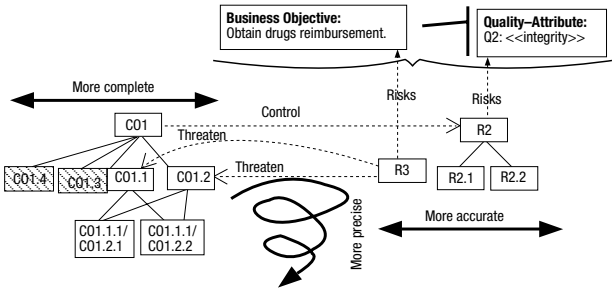
Figure 2—Control Objective Analysis



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 health care staff might deduce and expose an individual's disease. ⧫ Risk of compliance requirement

Control Objective:
CO1: Ensure that all data are complete and correct. Preserve ⧫ Q1 and Q2 from R1, R2 and R3
CO2: Hide personal information from the report. ⧫ Fulfill the compliance requirement risk.

Figure 3—Control Objective Refinement



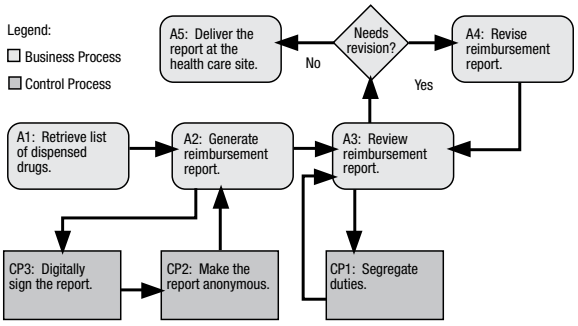
Risk:
R2: New <<fake>> drugs are added in the reimbursement report. ⧫ Risk to Q2 of business objective
R2.1: New drugs are added in the generated report. ⧫ Controlled by CO1.1
R2.2: New drugs are added while revising the report. ⧫ Controlled by CO1.2
R3: Collusion among actors to add new <<fake>> drugs in the report.

Control Objective:
CO1: Ensure that all data are complete and correct.
CO1.1: Ensure that 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 that 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 a case where a risk (R3) also threatens the effectiveness of control objectives (CO1.1 and CO1.2).

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 an 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 are in place, the challenge remains as to how they can be assessed and monitored in real time.

Figure 4—Interwoven Control Process and Business Process



	Control Processes
CO1.1.1	CP1: Segregate 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	CO3: 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…

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

For each control objective and process, analysts need to identify indicators that measure correctness and effectiveness. For these purposes, key assurance indicators (KAIs) and key security indicators (KSIs) are introduced:

- KAIs indicate the effectiveness of a control objective in assuring the compliance of business process. For example, to measure the assurance of CO1.1.1 (Ensure that all data are complete and correct), $KAI_{CO1.1.1}$, which measures how many times A2 and A3 are performed by the same actor, is introduced.
- KSIs concern the correctness of a control process in protecting the business process. For example, the control of CP1 (Segregate duties) 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, because business analysts are more concerned with the level of compliance 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 KAIs and KSIs are critical for monitoring, evaluating and improving the GRC implementation. The indicators are computed independently to distinguish between cases in which the KAI of a control objective is "low" but the KSI's associated control processes are "high." In the former case, analysts might conclude that there are some risks that have not 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 deployed controls.

IMPLEMENTATION GUIDANCE

To implement control processes and indicators in an SOA environment, one needs to specify which service events 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. The overall implementation of control processes and indicators is depicted in **figure 5**. In the previous 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, `AnonymizerService` for CP2 and `SoDService` for CP1. Essentially, there are two ways a control service works:

- Filters in/out a request to a business service.
- Verifies 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],[10] Enterprise Service Bus [ESB][11]). The ESB has the 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 *remove* 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 is 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 the results after performing some operations (e.g., *modify/add/remove* some data items, attach signature) or *remove* the result if it violates some policy (e.g., not sending confidential data).

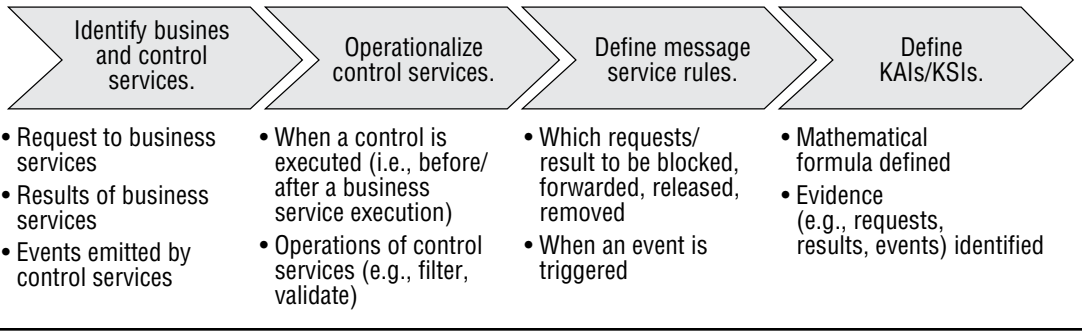
Besides implementing control processes, designers need to define the events (e.g., a service start/finish/suspend or messages exchanged among services) that will compute the KAI and KSI. To process these events, business activity monitoring (BAM)[12] can be used, since it allows one to analyze real-time 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**, a set of policies is specified to govern the actions of ESB and BAM.

ESB-related policies:

- **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 `AnonymizerService`.
- **Remove** the results of `GeneratorService` when there is no *release event* from `AnonymizerService` after four hours.
- **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.

Figure 5—Implementation Guidance



- **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-related policies (for CO1.1 [Ensure A2 and A3 are performed by different actors]):

- **KAI**—How many times has the same actor performed A2 and A3? Count how many times when the requester field of `ReviewService` request, which 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 the request comes from the A2 performer:

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

N_{delete} equals how many times the `SoDService` emits a *delete event*. $N_{same\text{-}req}$ equals how many times the requester field of `GeneratorService` request is the same as that of the `ReviewService` request.

CONCLUSION

The MASTER methodology, with its related set of tools, 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 support for integration and interoperability of legacy systems and applications developed by various vendors. The MASTER methodology allows one to control the execution flow of the 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.

AUTHOR'S NOTE

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

ENDNOTES

[1] Fight Fraud America, *www.fightfraudamerica.com*
[2] Rasmussen, M.; "Trends 2007: Governance, Risk and Compliance: Organizations Are Motivated to Formalize a Federated GRC Process," Forrester Research, April 2007
[3] SAP Community, "SAP and Governance, Risk and Compliance (GRC)," *https://www.sdn.sap.com/irj/bpx/grc*
[4] Vecchio, Dale; "Leverage Your Mainframe Application With SOA," Gartner Research, October 2005
[5] Seeley, R.; "Forrester Sees Convergence of SOA and BPM," SearchWebServices.com, 9 January 2007, *http://searchsoa.techtarget.com/news/article/0,289142,sid26_gci1238154,00.html*
[6] Lotz, V.; E. Pigout; P.M. Fischer; D. Kossmann; F. Massacci; A. Pretschner; "Towards Systematic Achievement of Compliance in Service-oriented Architectures: The MASTER Approach," *Wirtschaftsinformatik*, 50(5): 383-391, 2008
[7] EU 7th Research Framework Program, MASTER: Managing Assurance, Security and Trust for sERvices, *www.master-fp7.eu/index.php?option=com_docman&task=doc_details&gid=16&Itemid=60*
[8] Deming, W.E.; "Out of the Crisis," MIT Press, USA, 2000
[9] This case study has been provided by Hospital San Raffaele Foundation. Its complete description is available at *www.master-fp7.eu/index.php?option=com_docman&task=doc_details&gid=53&Itemid=60.*
[10] Sun Microsystems, "Java Message Service," *http://java.sun.com/products/jms/*
[11] Richards, Mark; "The Role of the Enterprise Service Bus," C4Media Inc., 23 October 2006, *www.infoq.com/presentations/Enterprise-Service-Bus*
[12] Kochar, Harpal; "Business Activity Monitoring and Business Intelligence," *ebizQ*, 25 December 2005, *www.ebizq.net/topics/bam/features/6596.html*