Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2008 Proceedings | Mediterranean Conference on Information Systems (MCIS)

10-2008

REQUIREMENT ANALYSIS FOR PROCESS-CENTRIC CONTINUOUS MONITORING

Sofiane Sahraoui
American University of Sharjah, PO Box 26666, Sharjah, United Arab Emirates, ssahraoui@aus.edu

Munir Majdalawieh
American University of Sharjah, Sharjah, United Arab Emirates, mmajdalawieh@aus.edu

Reza Barkhi
American University of Sharjah, Sharjah, United Arab Emirates, mmajdalawieh@aus.edu

REQUIREMENT ANALYSIS FOR PROCESS-CENTRIC CONTINUOUS MONITORING

Sahraoui, Sofiane American University of Sharjah, PO Box 26666, Sharjah, United Arab Emirates ssahraoui@aus.edu

Majdalawieh, Munir American University of Sharjah, PO Box 26666, Sharjah, United Arab Emirates mmajdalawieh@aus.edu

Barkhi, Reza American University of Sharjah, PO Box 26666, Sharjah, United Arab Emirates rbarkhi@aus.edu

Abstract

With the emergence of mission-critical real-time systems becoming ever more important to the competitive strategies of corporations and their e-business and supply-chain models, an increasing number of process controls are being embedded into information systems, and co-processed with business transaction thus providing for the continuous monitoring of business operations. A parallel trend in the auditing industry is towards continuous auditing, able to provide management with real-time auditing of the functioning of controls and of business transactions, thus enhancing significantly management's ability to ensure compliance and make key business decisions. Continuous auditing requires that information systems are developed not only to fulfill business requirements but also continuous monitoring of transactions and other compliance and control requirements. This integration of business systems and their controls within a process-centric logic necessitates a likewise integration of their development processes. Subsequently existing tools and techniques for requirements analysis need to be recast within a hybrid and integrated approach dubbed requirement analysis for process-centric continuous monitoring or RA-PCCM, which consists of the concurrent analysis of operational systems, information systems, the control system, and the management system. Whilst efforts exist within the auditing community to outline a process-driven methodology for developing continuous auditing systems, this paper argues for integrating control development for continuous monitoring within the fold of information system development, hence restricting auditors to control monitoring assurance.

Keywords: Continuous auditing, continuous monitoring, requirements analysis, systems development

1. INTRODUCTION

Abundant literature has surged as of late to strongly advocate the concept of continuous auditing which has been rendered all the more urgent by new auditing compliance requirements (Coderre 2005), the fading of the traditional audit trail (Flowerday & von Solms 2005; Weiss 1980), the emergence of ERP systems and other highly integrated real-time enterprise computing platforms (cf. Woodroof & Searcy 2001; Li et al. 2007), and more generally the necessity to accompany the 'electronization of business' (Onions 2003) and the coming of the 'on-demand enterprise' (Nagaratnam et al. 2005). Continuous auditing requires strong controls and continuous monitoring of all transactions rather than *ex-post* control and auditing of a sample of transactions (Flowerday & von Solms 2005; Rezaee et al. 2002; ISACA 2002; Searcy et al. 2002). This paradigm shift in the area of auditing and internal control, necessitating a closer alignment of monitoring and auditing with business processing, is still struggling to take root in organizational and auditing practice (Heffes 2006; ISACA 2002). Chief amongst the factors that can be ascribed to this is the resistance to change of auditors accustomed to *ex-post* and transaction-based, though computer enabled auditing (Heffes 2006); the prohibitive cost of continuous auditing though the latter has also been associated with cost-saving for repetitious reporting (cf. Li et al. 2007); the significant knowledge and skill retooling of auditors, necessary to operate within an IT-infused continuous monitoring environment (Searcy & Woodroof 2003; Searcy et al. 2002), the non-readiness of organizations from either a maturity or technological level (ISACA 2002), but also and central to developments in this paper, the lack of a business-driven and integrated methodology for the development of continuous monitoring systems. Whilst the latter has already been advocated in the literature (cf. Nagaratnam et al. 2005; Flowerday & von Solms 2005; Kokolakis et al. 2000), its direct connection with information system (IS) development has been so far evaded. Carnaghan (2006) has nonetheless suggested key steps involving the use of process modeling tools and conventions in the design of continuous auditing systems; albeit within a traditional auditing approach premised on a disjunction between IS development and auditing, hence monitoring.

This paper will propose an integration of the analysis of controls within requirement analysis for process-centric continuous monitoring (RA-PCCM). RA-PCCM includes not only information and control system analysis but also analysis of the operational and management systems as well. Hence, the conventional IS analysis process will be augmented to include both process and control analysis within a comprehensive framework that consists of the concurrent development of the management, operational, control, and information systems. The argument for reintegrating business and monitoring requirements of IS development is based on the premise that controls are primarily meant to validate business requirements determination prior to auditing them. Continuous monitoring would hence validate business processing in its management, information, and operational aspects whereas continuous auditing would be targeted at control monitoring assurance (Coderre 2005) hence providing an element of external validation for the process-centric systems at work. The rationale for an overarching process-centric perspective is highlighted in section 2 of this paper and illustrated graphically in figure 1.

Section 3 will review the main features of the disjoint processes of IS development and control system development and suggest a framework for a comprehensive and concurrent development process yielding both IS requirements and their controls, namely RA-PCCM. Operational and management system development will not be explicitly tackled as there is ample literature to indicate how their respective developments have been integrated with that of IS and vice-versa (cf. Owen & Raj, 2003).

2. PROCESS-CENTRIC PERSPECTIVE ON INFORMATION PROCESSING AND CONTROL

A standard narrative about a procurement system will be used in this paper to illustrate the proposed RA-PCCM. It is the basis for the interactions shown in the figure below between the four systems: management, operational, control, and information systems. The essence of a process-centric approach is the organic intertwining of different systems in performing basic business process functions. Hence the business process is posited as the most natural venue for the modeling and design of assurance controls (Sadiq et al. 2007). An enriched description of the business process with an emphasis on its IS and control dimensions is provided below:

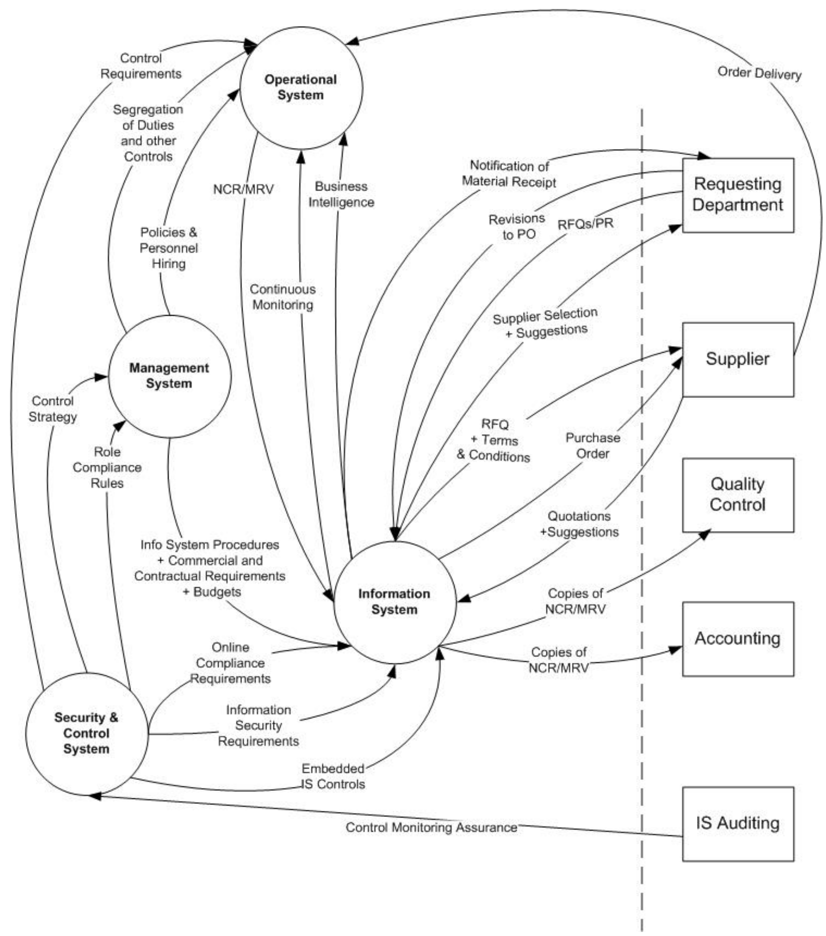

Figure 1: Process Centric View of Information Processing and Controls

As shown above, the four systems interact amongst themselves and with external entities at the same time. External entities are providers and recipients of net inputs and outputs to and from the business system respectively, which in a procurement process includes suppliers, the requesting department, quality control, accounting and IS auditing. The central role played by the IS in the business system derives from the logical nature of the model above as physical aspects of the system were not included. A physical representation of the business system would assuredly give more prominence to the role of the operational system. Given that management and control processes are primarily information-based logical processes, they should not be altered dramatically if one were to shift to a physical representation of the business system. The reason for choosing a logical rather than a physical representation of the business system derives from the objective of developing IS and their controls within an existing business system configuration. However were the objective to change and include a reengineering of the operational system as a primary focus of the development process, then a physical model would be more appropriate. In any case, both logical and physical models of the business system could be drawn to better comprehend the intricacies of interactions between the different systems. Such interactions are briefly described below:

As its name indicates, the operational system performs the basic operations of the organization or business process depending on the scope of analysis. Upon delivery of items by suppliers, a number of controls are executed to ensure that the items and their delivery process fulfill the objectives of the procurement process which can be expressed in terms of authenticating suppliers, checking item physical conformity, validating delivery conditions, etc. Such controls are set up a priori and are systematically executed upon each delivery. The management system provides the basic policies, structure, and routines within which procurement takes place. Depending on the degree of compliance sought, the control system will ensure that the operational system moves within the sphere delineated by the management system. Finally, the IS will provide the institutional memory and intelligence dimensions for the other systems. It will first record the delivery transaction executed by the operational system and hence provide it with access to a number of controls that are executed in real-time. Such access is systematic as controls will be embedded in the TPS execution routines through database integrity constraints thus becoming part and parcel of the IS itself (cf. Li et al. 2007). Based upon a satisfactory execution of controls, the transaction will be validated and recorded in the appropriate repository. Such a satisfactory closure of the transaction is tributary to a number of exchanges between the IS and external entities including the input of a request for quotations (RFQ), the issuance of a purchase order (PO), the submission of quotations, the selection of suppliers, and finally the notification of material receipt. A number of controls are also executed during each of these exchanges as they were equally embedded in the IS by the control system, hence representing a dual interaction between the IS and the control system.

The intelligence dimension of the IS, as mentioned above, translates into on-line accessibility of processed information (that is beyond transactional data) which allows the operational system to function in a flexible manner. For instance, in case of non-conformity of items delivered with the approved order, the operational system might still proceed with the reception of the items and their storage in inventory if the supplier has been given a premium status based on its peculiar history and relationship with the organization. Such a decision could be taken by an empowered employee following a specific policy enunciated by the management system, or derive from an IS control. Conversely, the control system could trigger an alarm concerning the delivery of items whenever exceptions to auditor-defined rules are detected (cf. Woodroof & Searcy 2001).

Whilst the above illustrates how process controls are enforced during the execution of transactions, and the ensuing interactions between the four systems that take place as a result, compliance can also be integrated into the inner workings of the other systems. The control system will define specific roles and responsibilities for the management system in ensuring compliance and a subsequent set of compliance requirements will be defined and embedded into the IS. Contrary to process controls which ensure the integrity of business transactions, embedded compliance requirements work in parallel with the intelligence dimension of the IS to generate real-time information about any potential violation of compliance, thus serving as an early warning system. This is part of continuous monitoring. When integrated with executive support systems that serve as corporate dashboards, compliance monitoring could thwart the type of violations that could go unchecked for a long time, thus enacting the vision of continuous auditing (cf. Searcy & Woodroof 2003; ISACA 2002). Finally, one key external entity that interacts with the control system is the IS auditing function whose role would be limited to control monitoring assurance (cf. Coderre 2005). This would happen however only within a scenario of pervasive continuous auditing; a situation that remains largely beyond reach for the time being as few organizations have yet realized the full vision and potential of continuous auditing (Heffes 2006). Moreover and without the development of tools and methods for the integration of information systems development with control system development, the ideal scenario of continuous auditing will further remain unachievable.

3. TOWARDS AN INTEGRATED SYSTEM DEVELOPMENT APPROACH

Traditionally the development of IS to support business process requirements on one hand and the development of system controls to ensure that IS fulfill their role on the other hand have been considered as distinct activities formulated by different stakeholders and having different lifecycles (Sadiq et al. 2007). The two development processes served the needs of different clients; business users for IS and auditors for system controls. The rationale for disentangling control from system development itself obeys a conventional auditing logic wherein control and the object of control should be dealt with separately (Grabski et al. 1987). However, periodical audits are challenged by the ever increasing changes introduced in IS (Wand & Weber 1989) as users' ability to manage IS-related business risk, is enhanced as a result of the processing and information reporting capabilities of ubiquitous IT infrastructures. The enhancement goes beyond traditional financial and compliance control to encompass management control which targets primarily the efficiency and effectiveness of both governance structures and embedded business processes (cf. Li et al. 2007). The scope of internal auditing is shifting as a result from compliance with controls and regulations to the improved efficiency and effectiveness of operations in the organization (Coderre 2005) prompting companies "to take a strategic view of compliance by adopting a *process-centric-approach*, with a focus on rationalizing their internal controls to ensure that the control framework is aligned with the key business drivers" (Li et al. 2007, p.432, *italics added*).

The shift of planning and analysis of IS from the exclusive attainment of business objectives to a dual focus on achieving objectives and controlling that such objectives are met on a continuous basis entails embedding risk assessment analysis and subsequent control design within the framework of business requirements analysis, such that continuous monitoring of business processes through IS-embedded controls is performed systematically. Within such an approach, key risks and controls are audited on a real-time basis (Heffes 2006).

Before RA-PCCM is illustrated, it is useful to briefly review the traditional approaches to IS and control system development.

3.1. Information systems development

IS development is invariably defined around the core concept of the system development life cycle or SDLC (cf. Hoffer et al. 2008). With the advent of prototyping as an alternative and iterative approach for IS development, a flurry of research and professional literature emerged to benchmark prototyping against the SDLC (cf. Mahmood 1987) despite the fact that prototyping itself encompasses the different stages of the SDLC albeit in an iterative manner. Notwithstanding, IS development comprises a number of phases invariably including planning, analysis, design, and implementation. The basic set of phases is augmented through further refinements of more detailed tasks pertaining to major phases. Hence, system analysis is sometimes decomposed into information requirements determination and subsequent requirements structuring or modeling whereas design is either logical or physical. It is beyond the scope of this paper to outline a comprehensive model of IS development but to simply stress its universal features, best illustrated through the SDLC concept.

Further technological developments led to the emergence of agile methods which ironically seemed to go back to the data processing model of EDP development wherein the focus of development was not as much on planning and analysis as it is on design and implementation (Sahraoui et al. 2007). However this is not due to a resurgence of any form of technical rationalism among system developers (cf. Hirschheim & Klein 1994) but rather to the peculiar format of open source software development which is predominantly held in the virtual environment of Internet working groups. Indeed agile methods were mainly crafted to fit the peculiar format of Internet-based open source software development characterized by an almost limitless access to design and programming resources (Feller & Fitzgerald 2002). Through a large number of trials and errors, valid designs could be found without undue focus on either planning or analysis. Contrary to prototyping, agile methods seem to provide a new paradigm for IS development, one that is centered on the latter stages of the SDLC. For purposes of further developments in this paper, we will not consider agile methods as they do not emphasize either planning or analysis. This can be justified on grounds that agile methods are more suitable for the development of software rather than complex business systems. By their very nature, business systems are embedded within their host organization and do not lend themselves to agile modes of developments. Only their software constituent parts could be developed through agile methods. Web Services, and XBRL for instance are open source technologies, developed through agile methods, that are becoming increasingly popular in the development of integrated systems with embedded audit modules (cf. Alles et al. 2006; Li et al. 2007; Flowerday & von Solms 2005; Onions 2003; Rezaee et al. 2002).

3.2. Development of process controls

With respect to the development of controls, these have been generally considered from the standpoint of risk analysis and security. Therefore controls were sought to lessen the risks that are intrinsic to the IS itself rather than its fulfillment of business objectives. Moreover there has been little in the way of defining process-based information requirements for purposes of risk analysis (Carnaghan, 2006), hence dissociating IS development from the development of the control system. As Kokolakis et al. (2000) put it:

"RA [risk assessment] methods rely on a very simplistic model of the organization defined in terms of assets, mainly data, hardware, and software, and subjects that may or may not have access to assets. *RA does not involve an analysis of the organization*." [p. 108, *emphasis added*].

The requirement of compliance for regulatory reasons also brought a second imperative to the development of controls. The situation has been exacerbated by forceful legislation like the *Sarbanes-Oxley Act* of 2002 and the ensuing IT control objectives (IT Governance Institute, 2007). Both imperatives of mitigating risk in its traditional (accounting) sense and ensuring compliance led to a rethinking of the disaggregation between IS development and control design. Controls had been hitherto defined in reaction to the implementation of specific IS operations rather than through a systematic development process at par with the development of IS themselves.

Moreover, the development of IS on one hand and controls on the other hand fell to different actors, namely system developers and auditors respectively. Despite early calls for auditor involvement in IS development to align controls with system objectives (cf. Grabski et al. 1987; Cash et al. 1977), and foster beneficial knowledge spillover between IS and control design (cf. Simunic 1984), developments in IT took auditors by a storm (cf. Searcy & Woodroof 2003). The latter would generally cling onto conventional risk analysis methods whilst IS evolved into modern days real-time systems that are more befitting of the continuous auditing paradigm (cf. Searcey & Woodroof 2003). Furthermore, this dualism in the development of IS and assorted controls is thought to be very costly as some have estimated the average cost of installing a control after IS implementation rather than during design to be between 10 to 20 times higher (Weiss 1980; Rittenberg & Purdy 1978). Though controls have become more sophisticated to match the likewise complexity of IS themselves, the adaptation of audit procedures to match control changes becomes a fruitless endeavor if controls are not planned and designed appropriately (cf. Wand & Weber 1989), which would suggest that auditors should be involved early on alongside IS developers to design appropriate controls that can be securely audited at a later stage (cf. Cash et al. 1977). Some further suggested that the lack of auditor involvement in the early development stages of IS led to a paucity of methods and approaches in analyzing systems control requirements and a subsequent inability of auditors to appropriately plan and conduct audits (Carnaghan 2006). Within a continuous auditing perspective however, the primary responsibility for planning, designing, and implementing controls is not the auditor's responsibility but management's:

Ideally, internal auditing is not part of the controls monitoring process and does not design or maintain the controls, thereby retaining its independence (Coderre, 2005, p. 2).

Another reason for the non-involvement of auditors in the development of controls alongside IS developers is the conflicting value-orientations of both communities. On the academic front for instance, the two communities are rather separate groups with different research agendas and methodologies (zur Muehlen & Rosemann, 2005). As professional communities, Rittenberg and Purdy (1978) relayed the dominant perception of auditors as being content with criticizing system weaknesses after installation rather than getting involved in system development:

To maintain independence, auditors will avoid situations (e.g. designing new systems) where they might have to audit their own work….They [auditors] evaluate the effectiveness of controls but are reluctant to assist data processing in designing needed controls (pp. 49, 51)

Whilst continuous auditing is not expected to alter the conflicting values and interests between business users and auditors, the latter are being called upon to develop a sound understanding of IS and the underlying information and IT architectures (Carnaghan 2006). Conversely, systems analysts are increasingly tasked with developing IS controls and were even hailed to be no less skilled in the design of a well controlled system (Grabski et al. 1987).

3.3. Requirement Analysis for Process-Centric Continuous Monitoring

The business process serves as a vehicle for connecting the different development processes (cf. Sadiq 2007). The management system yields the necessary plans and procedures for developing first the IS and secondly the operational system. The control system in turn ensures that the concurrent development of the information and operational systems converge towards the fulfillment of the objectives and compliance requirements enunciated by the management system. This is done by specifying a set of control goals to be fulfilled by the other systems along with control monitoring assurance criteria (See Figure 2). The continuous alignment and re-alignment of the four sub-systems is done through integrated process-centric analysis and design.

Analysis and design for process-centric continuous monitoring is posited as a comprehensive process integrating the development processes of four main systems, namely the operational system, the IS, the control system, and the management system. This does not imply that all of these systems would have to be rebuilt every time there is an IS development project taking place, for most of these systems or portions thereof would have been in place by the time a IS development effort is started. Only in situations where a new organization is being set up or a major change taking place, as in reengineering, would the possibility of concurrently building all four systems become concrete. In most other situations, the development effort will alter a pre-existing configuration of the existing systems. Hence analysis and design is about analyzing and designing a new configuration of the four systems in order to adjust to changes in any of them or in a combination thereof. This implies that not only controls will be analyzed and designed concurrently with the IS, but also appropriate changes to the operational and management systems will ensure that the new configuration is co-aligned, thus reducing the risk of imbalances being introduced as a result of a non-integrated development process. Concretely, this means that controls, applicable and later audited at the IS level, obey an operational logic primarily driven by the imperatives of efficiency and effectiveness, along with a management logic ensuring compliance. Figure 1 above illustrated a particular configuration of the four systems for a procurement process.

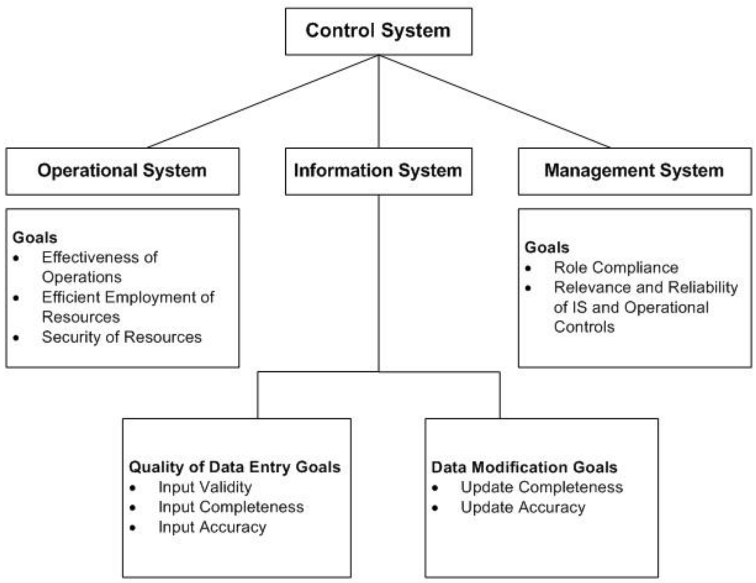

Figure 2: Generic System Control Goals

Rather than examining the substance and outcomes of controls themselves, “system evaluation by auditors is shifting to proving assurance of the integrity of the underlying system” (Dunn et al., 2005 p.84) and the new audit model is shifting accordingly from substantive test-based evaluation to a process-based one (Searcy et al. 2002). This is the norm nowadays for quality assurance. The ISO nomenclature for instance obeys this logic (cf. Carnaghan 2006). However and until continuous monitoring is fully functional, auditing will keep interfering with monitoring itself, yet remain a separate function conducted by independent auditors whose primary role is to ensure the relevance and reliability of existing business and compliance controls (cf. Coderre 2005).

Though controls could form the basis for subsequent audits, they are primarily targeted at ensuring the organizational validity of IS (cf. Markus & Robey 1983). The concurrent development of IS and their controls ensures the latter’s development from the perspective of users as part of the information requirement determination process, and their automatic update whenever changes to the underlying processes occur. Indeed and within a traditional approach where business objectives and controls are dissociated, it is difficult to pinpoint where control and auditing changes need to be made when the underlying business logic is altered (cf. Wand & Weber 1989). A control-driven IS development also redirects the focus of control from an external perspective to an internal one, thus raising user awareness about the importance of control as key in the validity of organizational systems.

4. CONCLUSION

The emergence of continuous auditing as both a bridging concept and practice between the fields of information systems and auditing affords IS professionals and academics a great opportunity to develop and deliver new tools and methods that fulfill the requirement of continuous monitoring. However it also represents a threat given the inroads of auditing into the IS area through efforts to develop SDLC-like processes for continuous auditing purposes, de facto making IS development efforts redundant. In line with the vision of continuous auditing as being restricted to control monitoring assurance, we have argued in this paper that IS development and control development should be integrated within systems development for process-centric continuous monitoring. The lack of an established approach and methodologies for the concurrent development of IS and the control system will however hamper efforts to gain control of the continuous monitoring agenda. We have attempted a modest effort in this paper by proposing an integration of IS and control system development, illustrating its application for requirement analysis. Though conceptual developments in this paper have been limited to requirements analysis, they could easily be extended to cover system design and implementation as well. The approach used in this paper was simply illustrative and not exhaustive in any ways. Proposals for the use of different modeling and design techniques to support a tighter connection between process-centric systems have been advocated before (cf. Kokolakis et al. 2000; Carnaghan 2006). Software engineering developments are likewise converging towards automated environments that could ultimately provide an integrated development environment for system development for process-centric continuous monitoring; chief amongst these is BPMN, an emerging business process modeling standard (cf. Shen et al. 2004, Owen & Raj 2003).

Beyond the theoretical argument, this reintegration has important practical fallouts, chief amongst these is the cross-fertilization of each of the analysis processes with the tools and methods of the other (i.e. IS requirement determination and control analysis). This cross use of tools and methods that has been hitherto restricted to one development process or the other will create a synergy effect within an integrated RA-PCCM.

References

Alles, M., Brennan, G., and Kogan, A. (2006). Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. International Journal of Accounting Information Systems, pp. 137-161.

Carnaghan, C. (2006). Business process modeling approaches in the context of process level audit risk assessment: An analysis and comparison International Journal of Accounting Information Systems, pp. 170-204.

Cash, J., Bailey, A., and Whinston, A. (1977). A survey of Techniques for Auditing EDP-Based Accounting Information Systems. The Accounting Review (4, October), pp.813-829.

Coderre, D. (2005). Global Technology Audit Guide --Continuous Auditing: Implication for Assurance, Monitoring, and Risk Assessment, The Institute of Internal Auditors, Florida, pp.1-34.

Dunn,C ., Gerard, F., and Grabski, S. (2005). Critical evaluation of conceptual data models. In International Journal of Accounting Information Systems, pp. 83-106.

Feller, J., and Fitzgerald, B. (2002). Understanding open source software development. UK: Addison Wesley.

Flowerday, S., and Solms, R. (2005). Continuous auditing: verifying information integrity and providing assurances for financial reports. Computer Fraud & Security, July, pp.12-15.

Grabski, S. V., Reneau, H.J., and West, S.G. (1987). The Comparison of Judgment, Skills, and Promoting Effects Between Auditors and system analyst. MIS Quarterly, June, pp. 151-159.

Heffes, E. (2006). Theory to practice; continuous auditing gains. Financial Executive; Sep, 22(7), p17-18.

Hoffer, J., George, J. and Valacich, J. (2008). Modern Systems Analysis and Design, fifth edition, Prentice-Hall.

Hirschheim, R. & Klein, H. (1994). Realizing Emancipatory Principles in Information Systems Development: The Case for ETHICS. MIS Quarterly, (18:1), pp. 83-109

ISACA Standards Board. (2002). Continuous Auditing: Is it a Fantasy or Reality. Information System Control (5).

IT Governance Institute, (2007). COBIT: Control Objectives for Information and Related Technology—Framework, Rolling Meadows, IL.

Li, Y., Roge, J., Rydl, L. and Hughes, J. (2007) Achieving Sarbanes-Oxley Compliance with XBRL bases ERP and Continuous Auditing" Issues in Information Systems (VIII: 2), pp.430-436.

Mahmood, M. (1987). System Development Methods-A Comparative Investigation. MIS Quarterly (11:3) September, pp. 293-311

Markus L. and Robey, D. (1983). The Organizational Validity of Management Information Systems. Human Relations; 36: 203-225

Zur Muehlen, M., and Rosemann, M. (2005). Integrating Risks in Business Process Models. In Australian Conference on Information Systems, Sydney, December.

Owen, R., and Raj, J. (2003). BPMN and Business Process Management: Introduction to the new business process modeling standard. Popkin Software: Accessed on-line in http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf

Rittenberg, L. V., and Purdy, C.R. (1978). The Internal Auditor's role in the MIS Developments. MIS Quarterly, December, pp.47-56.

Sadiq, S., Governatori, G., and Naimiri, K. (2007). Modeling Control Objectives for Business Process Compliance. In 5th International Conference on Business Process Management, Springer-Verlag.

Searcy, D., Woodroof, J., and Behn, B. (2002). Continuous Audit: The Motivations, Benefits, Problems, and Challenges Identified by Partners of a Big 4 Accounting Firm. In Proceedings of the 36th International Conference on System Sciences, The Computer Society.

Searcy, D., and Woodroof, J. (2003). Continuous Auditing: Leveraging Technology. The CPA Journal, May. Accessed on-line http://www.nysscpa.org/cpajournal/2003/0503/dept/d054603.htm.

Simunic, D. (1984). Auditing, Consulting, and Auditor Independence. Journal of Accounting Independence (22), Autumn, pp. 679-702.
Shen,H., Wall, B., Zaremba, M., Chen, Y., and Browne, J. (2004). Integration of business modeling methods for enterprise information system analysis and user requirements gathering. Computers in Industry (54), Elsevier, February, pp.307-323.
Sahraoui, S.; Al-Nahas, N.; Suleiman, R. (2007). An agile perspective on open source software engineering. In Handbook of research on open source software: technological, economic, and social perspectives / Kirk St.Amant, Brian Still, [editors], Hershey, PA : Information Science Reference.
Wand, Y., and Weber, R. (1989). A Model of Control and Audit Procedure Change in Evolving Data Processing Systems. The Accounting Review (64:1), January, pp 87-107.
Weiss, I. (1980). Auditability of Software: A Survey of Techniques and Cost. MIS Quarterly, December, pp. 39-50.
Woodroof, J., and Searcy, D. (2001). Continuous audit Model development and implementation within a debt covenant compliance domain. International Journal of Accounting Information Systems, Pergamon, pp. 169-191.