

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## An Aspect Architecture for Modeling Organizational Controls in Workflow Systems

Vu Nguyen, Ronald Lee and Kaushik Dutta  
Department of Decision Science and Information Systems,  
Florida International University, 11200 S.W. 8th Street, Miami, FL 33172, USA

---

**Abstract:** Besides business objectives, organizations have control goals concerning how these objectives are met. To achieve the control goals, organizations usually add controls into their business process. Over time, controls pile up and tangle with other activities; making it difficult to manage and reengineer evolving processes. In this study, we propose an architecture that enables dissection of controls from a process while preserving connections through a set of control points. The architecture aims to simplify the design process and minimize the effects of changes in controls.

**Key words:** Organizational control, internal control, business process, process modeling, workflow modeling

---

### INTRODUCTION

Business process has become a key concern of modern organizations to improve efficiency and effectiveness. A business process comprises a sequence of activities carried out by organizational actors to achieve a business goal. When a business process is automated in whole or in part, it becomes a workflow (Anonymous, 1999).

Process modeling captures various aspects of a business process and translates them into a process model that can be executed by workflow management systems. Besides business objectives, organizations have control goals with respect to reasonable assurance that the business objectives will be achieved and undesired events will be prevented or detected and corrected (Anonymous, 2005). To achieve the control goals, organizations add various control activities to their processes. These controls may have overlapping scope and as Lee (1998) observed “they grow more easily than they shrink”, making it difficult to design, implement and manage evolving processes.

In this study, we identify two aspects of a business process, the operational aspect and the control aspect. The operational aspect focuses on activities that collectively realize the business objective, while the control aspect concerns activities that direct the business process to its objective and ensure against deviations. We propose an architecture that represents the two aspects and their components. These aspects can be analyzed, designed and implemented separately and

linked together flexibly through a set of control points. The presented architecture is inspired by the concept of Aspect Oriented Programming (AOP) in software development (Kiezales *et al.*, 1997).

The aspect architecture enables decomposition of the whole process into operational and control aspects; thus simplifying the design process and utilizing expertise relating to each aspect. In addition, the process model can be stored in its respective aspects which are more common compared with the whole process; thus improving reuse of design components. The architecture separates control interface from its implementation so that changes of control do not affect the process structure. It also enables to detach and attach controls flexibly based on properties of processing cases.

### WHAT IS A CONTROL

**Organizational control:** Planning, organizing, leading and controlling are fundamental functions of management (Long *et al.*, 2004). Although control plays a critical role in organizational theory and practice, its implementation is still insufficiently studied as Oliver (1998) observed “the study of organizational control has a long history in administrative science and yet the need to examine the processes and implications of this phenomenon has never been greater”.

Sauermann (2004) identifies two perspectives of control in the organizational literature. First, control is considered as a process of directing, motivating and encouraging members of an organization to achieve the

organization's goal. This type of control is called "motivational control". Second, based on cybernetic view, control is defined as the "constant cyclic type activity of plan-do-compare-correct" with "its continuous, concomitant system of communication or flow of information" (Gigliani and Bedeian, 1974). The provision of information is critical to this type of control; it is so called "informational control".

Controls are distinguished primarily by control targets, influenced portions of an organizational production process, or by control systems, mechanisms that are used to govern organizational activities.

Organizations take inputs and transform to outputs by transformation processes. Distinguishing controls based on the portion of the process they influence, there are types of control: input control, process control and output control (Long *et al.*, 2004). Input controls, also labeled as feedforward controls or preventive controls, are designed to prevent errors from occurring. These controls are enforced at the input stage of the process. The objective of input controls is to anticipate and actively prevent problems, rather than passively react. Process controls, also referred to as throughput controls or concurrent controls, focus on the performance of the process. These controls monitor the process and correct the problem as they arise. Output controls, also called feedback controls or detective controls, are designed to detect errors after the process is complete and the outcome has been produced. The purpose of output controls is to check a completed activity and learn from mistakes.

Organizations achieve control goals through the use of various control systems. Three dominated control systems are market control system, clan control system and bureaucratic control system (Long *et al.*, 2004, Ouchi, 1979). Market control systems use external market mechanisms such as price competition and relative market share to establish standards used in the control systems. Clan control systems emphasize on the development and actualization of common values, traditions and beliefs and focuses on socializing members in ways that merge individual and organizational goals. Bureaucratic control systems use rules, regulation hierarchy lines of authorities and job specifications to direct subordinates in their tasks. This type of control systems emphasizes the use of "role" rather than individual people, the formal specification and the mechanical execution of business processes in an organization.

The focus of this study is on bureaucratic control systems in organizations and also referred in this paper as organizational controls, which may be defined as "the policies and procedures that help ensure that management's risk responses [selected relative to risk

assessment] are carried out. They include a range of activities-as diverse as approvals, authorizations, verifications, reconciliations, reviews of operating performance, security of assets and segregation of duties." (Anonymous, 2004).

Organizational controls are enforced at different conceptual levels of an information system (Schaad and Moffett, 2004). First, controls that are enforced through application logic and interface design; these controls are hard coded into application programs. For example, a customer cannot withdraw more than his available balance. Second, controls that are enforced by organizational structure and workflow design; these controls are specified by the separation of tasks, sequence of tasks, resource classification and resource allocation. An example of these controls is preparing order and approving order should be divided into two tasks and assigned to two different agents. Finally, controls that are enforced through audit and supervision; these controls are conducted without interference on any specific business activity. For instance, bank managers are notified of overdue payments.

With the growing application of workflows as a key tool for automating business processes, controls enforced at the workflow level have gained increasing concern. Besides business objectives, organizations seek to achieve control on how to get these objectives. A process model thus needs to reflect both the operational and control requirements.

**Control principles:** To achieve control over activities, most organizations lay down control principles and promulgate them though out the organization. These principles serve as the basis for designing and enforcing controls. Control principles are different from organization to organization due to its perception of risk and the requirement of compliance with its operating environment. Schaad and Moffett (2004) identify three most common principles:

**Separation of duties:** Separation of duties is a well-established principle of control firstly applied in financial organizations. The purpose of this principle is to prevent any one person from performing the whole process; thus reducing the risk of committing fraud on his own. This is achieved by breaking the process into at least two separate tasks and having these tasks performed by different people. Mistakes made by one person tend to be detected by others and frauds require collusion of at least two people (Fites *et al.*, 1989).

In the workflow context, Separation of duties has been divided into static separation of duties and dynamic separation of duties. The static separation of duties

enforces certain constraints during the build time of the workflow. An example of static separation of duties is the separation of tasks and assignment of these tasks into different people; so that a single person does not have access to all of these tasks. Dynamic separation of duties does not restrict the association of people and tasks at the build time; it however restricts the activation of tasks at run-time according to and the separation of duty specification. An example of dynamic separation of duties is a person has the right to do both preparing order and approving order, but he is not allowed to do these two activities on the same order.

**Delegation:** Delegation can be defined as the process in which a superior transfers to his subordinates the responsibility and authority for carrying out a particular task in certain conditions. In other words, delegation is the assignment of responsibility (i.e., delegation of obligation) together with appropriate power (i.e., delegation of authority) to carry out a task, usually in accordance with conditions or limitations prescribed for that delegation (e.g., being limited to a threshold amount or to a position area).

Delegation is an important mechanism for decentralizing management, especially in large organizations where it is impossible for a single person to manage all organizational activities. An example of delegation is a manager of a bank who allows tellers to complete withdrawal transactions less than \$1,000 without the manager's approval. In process modeling, delegation is enforced through the hierarchical structure of organizations.

**Supervision and review:** Supervision and review control whether a delegated task is done as required. Supervision is a process including various activities such as monitoring, evaluating and advising and typically carried out on subordinates by their direct supervisors. Review is a control activity which is usually carried on a specific activity and it does not necessarily need to be performed by a superior position. Both supervision and review can be represented by adding control tasks to the process.

### WORKFLOW REFERENCE MODEL

The Workflow Management Coalition (WFMC) defines workflow as “the automation of a business process in whole or in part, during which documents, information or tasks are passed from one participant to another according to a set of procedural rules” (Anonymous, 1999). A Workflow Management System (WFMS) is a system that defines, manages and executes the workflows.

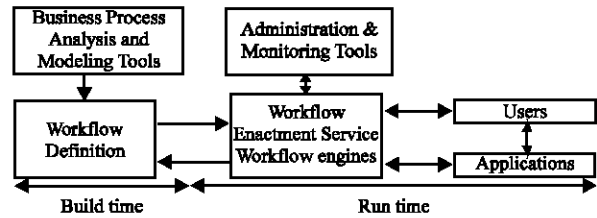


Fig. 1: Workflow Management Coalition (WFMC and) reference model

A key feature of workflow management systems is the separation between business process management and applications that support the execution of tasks in the process. Process logic is implemented in a workflow management system while application logic is implemented in application components.

The Workflow Reference Model proposed by WFMC describes a generic architecture including build time and run time functions (Fig. 1).

**Build-time functions:** The build-time functions concern with defining and modeling the workflow process and its constituent activities. During this phase, the real world business process is identified and translated into a workflow definition which can be enacted by workflow engines. The definition of workflows includes two main parts: the process model and the organization model. The process model comprises a number of tasks with associated computer and/or human operations and a set of rules that govern the execution of tasks. The organization model includes organizational objects such as roles, permissions and users and organizational relationships (i.e., organizational structure). Some workflow systems allow modification of process definitions at run-time, as indicated by the feed-back arrow.

During the modeling process, along with other business requirements, various controls are incorporated into the workflow definition. Examples of these controls are the approval of purchase orders, check of invoices against purchase orders and against goods received and so on.

**Run-time functions:** Workflow enactment service is the central component of a workflow system. It creates, manages and executes workflow instances through the use of one or more workflow engines. Workflow engines interpret the process definition, create and control instances of the process, schedule various activities of the process and invoke human or IT application resources necessary to these activities. The run-time functions link between the modelled process and the real

world process through the interaction with human or IT application resources.

At run-time, workflow enactment service enforces controls built in the workflow definition. In addition, it interfaces with a set of administration and monitoring tools to perform various functions for monitoring and controlling purposes, e.g., transactional control, resource control, process supervision, error handling, etc.

### THE ASPECT ARCHITECTURE FOR MODELING CONTROLS

**The overall architecture:** A business process is a set of related tasks performed to achieve a business goal, usually within the context of an organizational structure which defines functional roles and relationships (Anonymous, 1999). We identify two aspects of a business process, the operational aspect and control aspect.

The operational aspect focuses on the operational tasks, tasks that collectively realize the business objective and the agents who carry out these tasks. The control aspect focuses on tasks that aim to direct the business process to its designed objective and ensure against deviations. These tasks are usually carried out by controllers or supervisors in the organizations.

The operational and control aspects interact through two interfaces. The first interface relates control agents and operational agents. This relationship is represented in the organization model that describes the structure of the organization (organizational model). The second interface describes the relationship between control tasks and operational tasks. This relationship is depicted in the process model that comprises tasks and their dependencies (process model). Interface 3 and 4 link the organizational model and the process model; that is, they link tasks to agents who are eligible to perform them (Fig. 2).

Modeling controls in workflow systems involves identifying and representing components and relationships in each aspect, as well as the links between components of the two aspects. In this study, we focus on the second interface which describes the relationship between control tasks and operational tasks in a process model.

**Process architecture:** Within the context of process modeling, the operational aspect focuses on operational tasks and the sequence of these tasks without concern for control considerations and it focuses on the core process that does not include any control. On the other side, the control aspect concentrates to control activities which

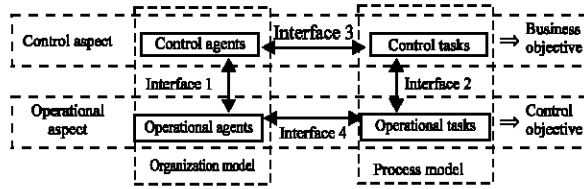


Fig. 2: Aspect architecture for modeling controls

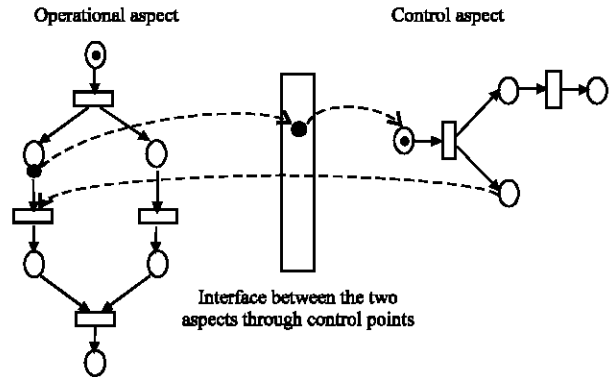


Fig. 3: Process architecture

comprise tasks that help to prevent, detect and correct problems.

To separate controls from the operational process we need to do two things. First, identifying points in the operational process where controls enforce, we call these points control points; second, establishing connections between the two aspects through control points (Fig. 3). These two problems are discussed in the following sections.

**Benefits of the architecture:** Separation of the operational and control aspects at the architecture level brings a number of benefits which can be summarized as follows:

- During the design, the operational and control aspects can be considered independently; thus making the design simple and accelerating the designing process.
- Each aspect provides a specific view corresponding to specific domain knowledge, the operational and control domains; therefore utilizing expertise in each domain. From the operational view, designers concentrate to the operational activities and the routing of these activities in order to obtain the business objective. From the control view, by contrast, controllers focus on control activities and do not necessarily need to know the structure of the operational process.

- A process model can be stored in its respective aspects, thus enabling reuse of design components. Processes without controls are very common for similar organizations. They tend to differ because of different controls added, particularly controls derived from perception of risks. Common operational processes can be stored as process templates which serve as skeletons for creating new processes. In addition, common controls can be saved as control patterns which can be reused across projects in an organization or across organizations.
- The architecture enables separation of control interface from its implementation, so changes of controls do not affect the operational process. Controls can be implemented as components linked with the operational process either through directly calling and providing services or through a service broker as in the Service Oriented Architecture (SOA).
- Last but not least, representing control and operational functions in separate aspects enables the potential automation of detaching and attaching controls from and to the operational process. Therefore, we can model flexible processes where different controls can be enforced in different cases. It also opens the ability to evaluate flexibly the performance of a process with or without a specific control.

**COMPONENTS OF THE OPERATIONAL ASPECT**

**Case:** Cases are entities of interest that are handled by the designed workflow. Examples of cases are insurance claims, purchase orders, visa applications, tax declarations, etc. A case has life cycle from the moment it enters the process to the time the process is complete(Aalst and Van Hee, 2002).

Sometimes cases are distinguished because they are objects of different controls. For instance, if there is a rule that every withdrawal of transactions more than \$5,000 need ands the supervisor’s approval, then we have to distinguish two types of cases: withdrawals that do not exceed \$5,000 and withdrawals that exceed \$5,000.

Similar cases are represented by a case type. A workflow usually handles one type of cases. To reduce explosive number of workflows, similar workflows can be combined into a composite workflow that can handle multiple case types.

**Operational task:** The basic component of a workflow is task. A task is a logical unit of work that is carried out as a single whole. As a single unit of work, a task is not

considered to be split into smaller tasks and is always carried out in full. If anything goes wrong during the execution of a task, we must return to the beginning of the entire task (Aalst and Van Hee, 2002).

We distinguish tasks to be operational tasks and control tasks. Operational tasks are those that collectively realize the business objective of the process. Operational tasks are critical to the process; that is, missing one of them, the process may not reach its designed objective. Examples of operational tasks are sending orders, receiving products, receiving invoices and paying bills.

**Routing:** In a process model, all possible execution paths are laid down at the build time. Routing specifies which tasks are carried out and in what order for specific types of cases; that is, it describes how cases are routed through a workflow. There are four basic routing constructions (Aalst and Van Hee, 2002; Jalonski and Bussker, 19996):

**Sequential routing:** enables a task in the workflow after the completion of another task in the same process. If task A and B are sequential, they may be neighboring or not. In Fig. 4a, task 1 and task 2 are sequential and neighboring. While task 1 and task 3 are sequential but not neighboring.

**Parallel routing:** allows more than one tasks execute at the same time in any order. A parallel routing starts with an AND-split and ends with an AND-join. The AND-split supports simultaneous execution by allowing single

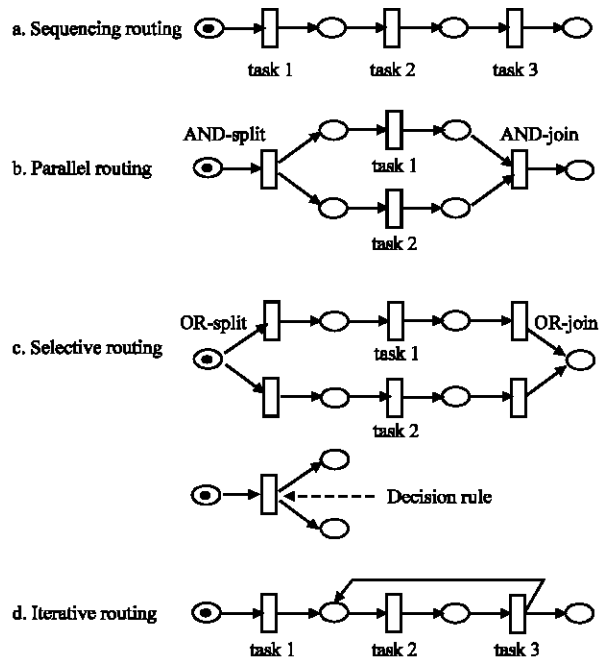


Fig. 4: Routing constructions

threads to be executed in parallel. The AND-join brings together multiple parallel threads into a single thread. In Fig. 4b, task 1 and task 2 can be carried out concurrently at any order.

**Selective routing:** represents alternative paths; each path handles a specific type of cases. At the run time, the workflow will choose one of the alternative paths for a given case. The selective routing comprises of an OR-split and an OR-join. The OR-split is a point where a single thread of control makes a decision upon which branch to take among multiple alternative workflow branches. If the choice between alternatives is nondeterministic, i.e., the choice is made by agents or factors outside the process model, it is represented arcs starting from a place. If the choice is deterministic, i.e., alternative is chosen based on properties of a case, it is represented by a task acting as a decision rule. This is analogous to the distinction made in decision trees, where the two types of branches are endogenous-made by us, or exogenous-made by external factors. The OR-join is a point where two or more alternative branches re-converge to a single activity.

**Iterative routing:** represents the repeated execution of tasks until certain results obtained. In Fig. 4d, task 3 is a control task that checks the result of task 2. Based on the result of this check, task 2 may be executed once more.

In a process model, the routing represents the logical dependencies among operational tasks, between operational tasks and control tasks and among control tasks. For instance sending order must be after preparing order; verifying order, if exist, must be after preparing order and before sending order; approving order, if exist, must be after verifying and before sending order; cross-checking between receiving invoice and goods can only be done after receiving both invoice and goods.

**Operational block:** A workflow block, implemented as a sub-workflow, is a group of tasks and their routing that can be isolated as an autonomous segment of a container workflow. When a block task is started, it passes control to the first task of the corresponding sub-workflow. This sub-workflow will pass control back to the block task upon completing its execution (Fig. 5).

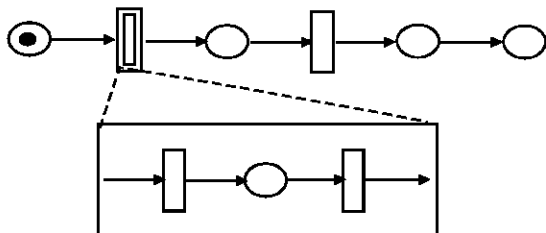


Fig. 5: Workflow block

Identifying and reusing recurring segments of workflow as blocks help to avoid redundant design, making the modeling process easier, faster and less prone to errors (Verginadis and Mentzas, 2004).

For the purpose of modeling controls, we consider a workflow block as an operational block if it does not contain any concerned controls in its own structure. When modeling controls, we treat an operational block as a black box; that is, we focus on the inputs and outputs of the block rather than of its tasks. For example, to model inter-organizational controls, we may consider a segment of workflow containing internal tasks as an operational block.

**Operational process template:** Operational process comprises operational tasks and the routing of work along these tasks. An operational process does not contain any control. An example of operational processes is a procurement process that consists of sending order, receiving goods, receiving invoice and making payment; no approval, verification or reconciliation presents in the process. The routing of tasks in an operational process is specified by operational constraints, i.e., the input-output dependencies among these tasks.

The operational process is the core of a business process and it is not affected by changes in controls. Organizations may have the same operational process but their business processes are different because of different controls added. Detaching controls from a process improves reusability of the process model.

Operational processes that are commonly used in whole or in part can be saved as operational process templates. These templates are used as skeletons for creating new processes.

**COMPONENTS OF THE CONTROL ASPECT**

**Control task:** Control activities are actions supported by policies and procedures that help to ensure management directives are carried out and necessary actions are taken to address risks (Anonymous, 2005). Control activities direct the business process to its business objective and help to prevent, detect and correct errors and frauds that may happen during the process execution. Controls are supporting activities and do not contribute to the “value-adding” of the process. Examples of control activities are approval, authorization, verification, reconciliation, performance review, security of assets and segregation of duties.

In workflow systems, control activities are built as part of the process, so called built-in controls, or enforced through various administration and monitoring tools, so called additive controls. For the

purpose of process modeling, we focus on controls being built into process models.

Control activities involve evaluating inputs or outputs of operational tasks against the pre-established standard and taking corrective actions when the standard is not met. We decompose control activities into control tasks and corrective tasks. A control task evaluates information received from one or more operational tasks. Depending on the result of this evaluation, appropriate corrective tasks are executed.

In a process model, controls are enforced through: (1) the separation of operational tasks, e.g., making payment and receiving goods should not be combined in to one task; (2) the incorporation of control tasks, e.g., there should be control task(s) for each operational task; and finally, (3) the appropriate order between operational tasks and control tasks, e.g., a preventive control must precede the operational task, a detective control must follow the operational task, a reconciliation must follow the merging of all involving tasks. The relationship between operational tasks and control tasks is represented by interface 2, Fig. 2.

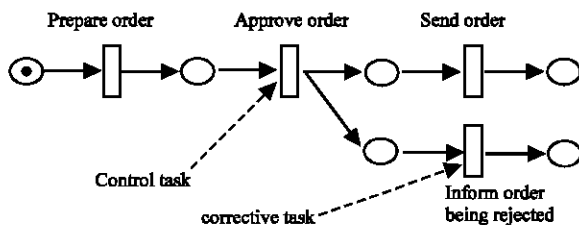
Another issue with modeling built-in controls involves the assignment of tasks to agents who are eligible to perform them. In process modeling, this is done through the classification of agents (interface 1, Fig. 2) and the allocation of tasks to agents (interface 3 and 4, Fig. 2).

**Corrective task:** Corrective tasks include actions being carried out after the control task is done and the standard is not met. These actions are categorized into immediate corrective actions, changing activity at once to get the performance back on track and basic corrective actions, determining why performance deviated and correct the source of deviation. Another type of corrective actions is revise the standard because it was set too high or low; these actions involve changes in the operational process.

In process modeling, we focus on immediate corrective actions. Example of these actions are: warning but continuing the process, warning and stopping the process, warning and waiting for decision on continuing or stopping the process, repeating the process in whole or in part until the standard is met, doing additional activities before repeating the process and so on.

In Fig. 6a, the control task approving order will check the order against available budget, funding limit, price list, etc. If every thing is OK, the order is approved; if not, the order is rejected and the reason for rejection is informed. In this example, approve order is a control task and inform order being rejected is a corrective task. In Fig. 6b,

a. Warning and terminating the process - represented by a deterministic choice



b. Repeating part of the process - represented by an iteration

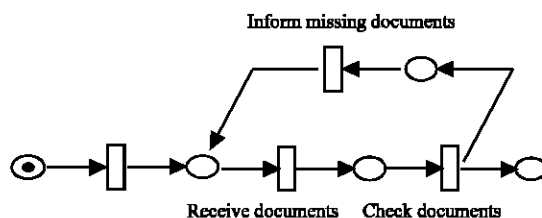


Fig. 6: Control patterns

application documents for issuing a visa will be checked. If all required documents are submitted, the application will be passed to the next step; if not, the applicant will be informed to correct or submit additional documents.

**Control pattern:** Chen (1992) defines control patterns as stereotypical relationship between agents, tasks and documents. In process modeling, control patterns comprise relationship between agents, tasks and cases. A control pattern can be represented by formal logic (Chen, 1992; Gligor *et al.*, 1998; Lee *et al.*, 2005), Petri Nets (Chen, 1992; Aalst *et al.*, 2003. Konstantin and Henrik, 2001), or Workflow Description Language (Jalonski and Bussker, 1996; Bussler, 1994).

Within the context of process modeling, a control pattern comprises the following components:

- Pre-conditions: Information for a control decision.
- Post-conditions: Control decision.
- Control task: Task that performs the checking and evaluating function.
- Corrective tasks: Tasks being carried out if the standard is not met.

Figure 6 shows two control patterns. In the first pattern and (Fig. 6a), pre-condition is the order prepared, post-condition is the order rejected or approved, control task is checking order against standard (e.g., budget, funding limit, price list, etc) and the corrective task is informing the rejection reason.

In the second pattern and (Fig. 6b), pre-condition is the documents received, post-condition is the documents



returned or accepted, control task is checking documents against standard (e.g., list of documents required, specifications of each documents, etc.) and the corrective task is informing missing or inappropriate documents.

**LINKING THE OPERATIONAL ASPECT AND CONTROL ASPECT**

**Control point:** Control points are positions along the operational process where a specific control can be attached. These points serve as the joint points between the operational and control aspects. A control point is represented by a dark circle right after the place that provides sufficient information for a control decision.

The introduction of control points enables the separation of controls from the operational process at both the build and run time. At the build time, controllers do not need to know the structure of the operational process. What they concern are a set of control points and the outcomes at these points. On the side, designers will specify points along the process where required information for a control can be provided but do not need to know how the control is performed. At the run time, a control point can serve as the interface of control while its implementation is stored in separate components.

Control points of a specific control are established depending on the logical order between the control and its corresponding operational task(s), the sufficiency of information for control decision, the effective scope and effective time of corrective activities and finally the logical correctness of the defined process. Attaching controls at different control points may affect the overall performance of the process, as being examined in the following examples.

A preventive control must be positioned before its corresponding operational task and should be located as early as possible in the process; however, it cannot be done before getting sufficient information, which is fed by preceding tasks, for making decision. To give an example, the process of issuing visa with three operational tasks: receiving application and supporting documents, scheduling an appointment and finally doing an interview. In this process, check of sufficient and appropriate documents is a preventive control of doing the interview. If this control is enforced, it must be done before the interview and after receiving the documents. To improve performance, in this case is to reduce unnecessary processing time, the control should be done right after receiving documents and before scheduling the interview. In this example, we have two control points for a specific control. Inserting control at different control points affects performance of the process (Fig. 7).

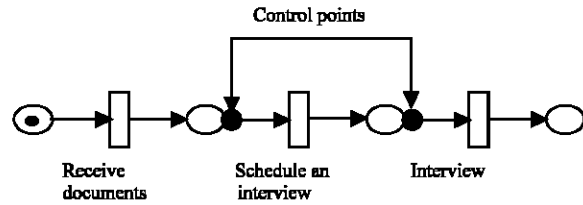


Fig. 7: Control points-example 1

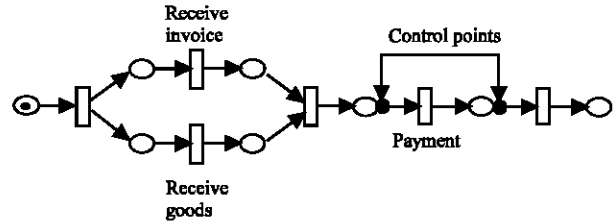


Fig. 8: Control points-example 2

Another example is checking of invoice against goods received. This detective control cannot be done before receiving of both invoice and goods and should be done before making payment; so that payment can be voided if and there is a mismatch between invoice and goods and is detected. This detective control can be done after making payment, but in this case we cannot stop payment even if significant problems are detected. Instead, other corrective actions could be done such as notifying the problems to vendors. In this example, enforcement of control at different control points affects the corrective actions (Fig. 8).

In the first example, a preventive control could be enforced after receiving documents and before doing the interview. We call this segment the valid segment of the control. A control can be applied at any points within its valid segment, still preserving logical correctness of the workflow.

If two tasks are sequential, the detective control of a task can function as the preventive control of the subsequent task. In the first example, checking documents can be considered as the detective control of receiving documents or as the preventive control of doing an interview. However if two tasks are not sequential, this statement may not be true. Look at the second example, the detective controls of receiving invoice and receiving goods cannot replace for the preventive control of payment which requires reconciliation between invoice and goods.

In an inter-organizational workflow, adding a preventive control before a public task (i.e., tasks that send or receive information from other workflows) could violate the logical correctness of a workflow-net (Aalst and Van Hee, 2002). In the flowing example, task 1 is a

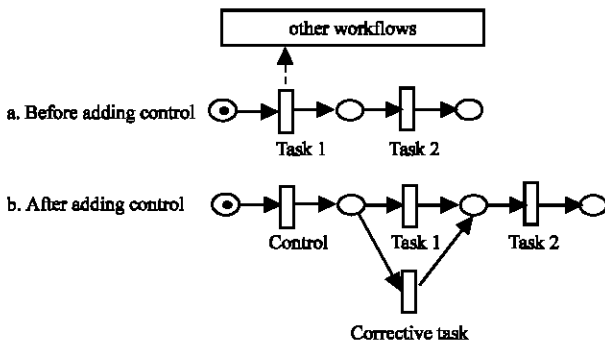


Fig. 9: Control points-example 3

public task that sends information to other workflows. Before adding control, task 1 is always executed; so that other workflows involving in the inter-organizational workflow receive information necessary for their process. By adding a preventive control that checks pre-conditions of task 1, the control may route the process to a corrective task based on the result of this check. Thus, there may be a potential execution where task 1 is not carried out (Fig. 9). Further discussion can be found by Aalst and Weske (2001).

**Linking the aspects:** The second problem involves establishing connections between the operational and control aspects through control points. These connections specify how a control is treated at both build and run times.

From the design standpoint, control points work as pre-conditions of a control activity. Each control activity can be considered as a workflow block consuming information provided at the control point and returning the execution to a specific task within the operational process. This task may be the ending task (i.e., the process is terminated), the next operational task (i.e., the process is continued), other control task (i.e., further control enforced), the beginning task or a preceding task (i.e., the process is repeated in whole or in part). The workflow needs to be verified after attaching or detaching controls to ensure the logical correctness (Aalst and Van Hee, 2002), especially with real-time collaborative workflows (Du and Jian, 2003).

From the implementation standpoint, control points serve as the interface between control tasks and the operational processes, while its content can be stored in separate components. The interface acts as a contract between operational processes and the control components so that workflows can be implemented without knowing details of the implementation of controls. This would enable the plug-and-play of controls dynamically during runtime without affecting any other

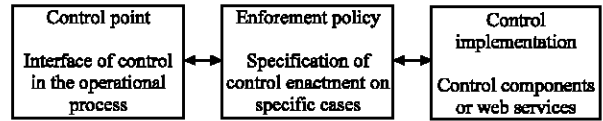


Fig. 10: Flexible enforcement of controls

parts of the process. For instance, we can have a control component that takes two documents, invoice and order and results true or false depending on whether the invoice matches the order or not. In this scenario, how these documents will be compared depends on the exact format of the two documents. If the organization migrates from one format of document to another, a new control component can be built and loaded to do such comparison, without affecting other parts of the process.

Another possible implementation solution is through web services, by which the workflow management systems do remote call ands to achieve particular control functionality (Fig. 10). The remote call locations are determined dynamically at runtime by parsing the WSDL (Web service definition language) file. In this scenario, organizations that use a control act as service requesters and organizations that carry out the control are service providers. Control functions are actually “outsourced” to external bodies such as audit institutions or government. This model is especially appropriate for small to medium enterprises due to their lack of resources and expertise to properly implement, manage and operate mandated control enforced by laws and regulations. In this model, however, there is possibility of “mass destruction” when a malicious control can demolish, or maybe more dangerous, manipulate all involved processes.

Separation of controls from operational processes enables flexible enforcement of control based on knowledge about the processing case at runtime. At a control point, the workflow management systems will check whether the control is necessary for the case or not. This is done through enforcement policy, a set rules that define which controls are enforced on which cases and in which conditions. Examples of such policies are: IF Value of the order is more than \$100 THEN manager’s approval is required and; and, IF vendor is A THEN payment must be done immediately upon receiving invoice.

**Putting together in an example:** Figure 11 shows a simple procurement process viewed from two different aspects. In this figure, the dash arrows represent the flow of process if controls are not executed.

The operational aspect comprises tasks that are critical for the process, including requesting purchase, preparing order, receiving goods and invoice and then

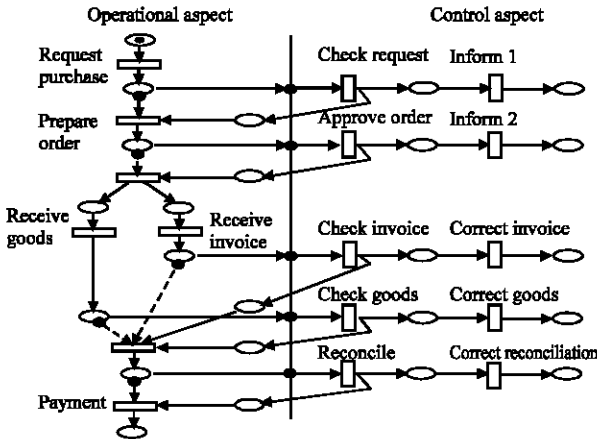


Fig. 11: Operational and control aspects

making payment. These tasks are common for most of the procurement processes and they are all necessary for the business objective, which is to get the goods as required.

As viewed from the control aspect, controls need to be enforced at many points along the process. For instance checking the request against the current inventory, the order against available budget, vendors and prices, the invoice and goods received against each other and again the order and so on. These controls may not be the same for all executions, e.g., payments to a vendor who has good reputation and well established relationship can be done before receiving goods or invoice.

**CONCLUSION AND FUTURE WORK**

Organizations seek to achieve business objectives as well as control objectives through the automation of business processes. Process modeling thus needs to reflect activities that support both objectives. Addressing organizational control in process modeling is not a new issue. However, research in this field rarely distinguishes between the operational aspect and control aspect of a process and little has been done on their relationship in supporting organizational goals.

In this study, we have identified the two aspects and proposed an architecture that enables the separation of these aspects. Each aspect can be analyzed, designed and implemented separately and linked together through a set of control points.

The proposed architecture provides a number of benefits in design and implementation of evolving business processes where changes in control do not affect the structure of the operational process. The study, however, is limited in its scope to the relationships between operational tasks and control tasks within a

process, while a thorough research on organizational control needs to examine a larger view including the assignment of tasks to agents and the relationship among agents, because at the end, tasks are carried out by agents. This limitation brings up many issues for future research, such as the delegation of obligation and authority or the contractual relationships among agents involved in a process (interface 1, Fig. 2) and the binding of tasks to agents in an organizational and inter-organizational process (interface 3 and 4, Fig. 2). These are all interesting areas for the application of deontic logic in process modeling (Lee, 1988).

Another limitation of this study is it clusters all controls to a single aspect, so it is unable to describe interaction among different control perspectives (e.g., internal control, external control). Having realized this shortcoming, we are currently elaborating an architecture that can represent multiple aspects of control. The new architecture would also be able to represent the N:M relationship between control activities and operational activities. That is, one control can supervise many operational activities; inversely, an operational activity can be managed by many controls. The enforcement of one control may affect a chain of activities, including operational activities and other control activities, scattering through out the process.

**REFERENCES**

Aalst, W.M.P. van der and K.M. van Hee, 2002. Workflow Management: Models, Methods and Systems. Cambridge: MIT Press.

Aalst, W.M.P., van der, A.H.M. ter Hofstede, B. Kiepuszewski and A.P. Barros, 2003. Workflow patterns. Distributed and Parallel Databases, 14: 5-51.

Aalst, W.M.P. van der and M. Weske, 2001. The P2P approach to inter-organizational workflows. Proceedings of the 13th International Conference on Advanced Information Systems Engineering. Lecture Notes in Computer Science. Springer-verlag, Berlin, 2068: 140-156.

Anonymous, 1999. Terminology and Glossary. Document Number WFMC-TC-1011. <http://www.wfmc.org>.

Anonymous, 2005. COBIT Framework-Control Objectives for Information and related Technology. [www.isaca.org](http://www.isaca.org).

Anonymous, 2004. Enterprise risk management-integrated Framework: Application Techniques, Coso, pp: 63.

Bussler, C., 1994. Policy resolution in workflow management systems. Digital Technical J., 6: 26-49.

- Chen, K.T., 1992. Schematic evaluation of internal accounting control systems. Ph.D Thesis, University of Texas at Austin.
- Du, Y. and C. Jiang, 2003. Towards a workflow model of real-time cooperative systems. Proceedings of the 5th International Conference on Formal Engineering Methods, ICFEM 2003, Singapore. Lecture Notes in Computer Science, 2885: 452-470.
- Fites, P.E., M.P.J. Kratz and A.F. Brebner, 1989. Control and Security of Computer Systems. Computer Science Press Inc.
- Gigliani, G.B. and A.G. Bedeain, 1974. A conspectus of management control theory: 1900-1972. Acad. Manag. J. 17: 292-305.
- Gligor, V.D., S.I. Gavrilă and D. Ferraiolo, 1998. On the formal definition of separation-of-duty policies and their composition. IEEE Symposium on Security and Privacy. Oakland, California.
- Jalonski, S. and C. Bussker, 1996. Workflow management: Modeling Concepts, Architecture and Implementation. International Thomson Computer Press, 1996.
- Kiczales, G., J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.M. Loingtier and J. Irwin, 1997. Aspect-oriented Programming. Proceedings European Conference on Object-Oriented Programming. Springer-Verlag, Berlin, Heidelberg and New York, 1241: 220-242.
- Konstantin, K. and S. Henrik, 2001. Modeling and analyzing separation of duties in petri nets workflows. Proceedings of the International Workshop on Information Assurance in Computer Networks: Methods, Models and Architectures for Network Security. Lecture Notes in Computer Science, 2052: 102-114.
- Lee, R.M., 1988. Bureaucracies as deontic systems. ACM Transactions on Office Information Systems, Special Issues on Language-action Perspectives, 6: 87-108.
- Long, C.P., S.B. Sitkin, L.B. Cardinal and R.M. Burton, 2004. An information processing model of organizational control: A computational model of system-level effects. <http://www.olin.wustl.edu/workingpapers/pdf/2004-10-001.pdf>.
- Lee, R.M., K. Henry and V. Nguyen, 2005. Toward open exchange of bureaucratic controls. The 2nd Workshop on Applications Of petri Nets to Coordination Workflow and Business Process management. A Satellite Event of the 26th International Conference on Application and Theory of Petri Nets and Other Models of Concurrency, Miami, Florida.
- Oliver, C., 1998. Critical theory perspectives on control. Administrative science quarterly, 43: preface.
- Ouchi, W.G., 1979. A conceptual framework for the design of organizational control mechanism. Manag. Sci., 25: 833-848.
- Schaad, A. and J. Moffett, 2004. Separation, review and supervision controls in the context of credit application process-a case study of organisational control principles. Proceedings of the 2004 ACM Symposium on Applied Computing. Nicosia, Cyprus.
- Sauermann, H., 2004. Two Sides of the Same Coin-An Integrated Model of Motivational and Informational Control Processes. [http://www.casos.cs.cmu.edu/events/conferences/2004/2004\\_proceedings/Sauermann\\_Henry1.doc](http://www.casos.cs.cmu.edu/events/conferences/2004/2004_proceedings/Sauermann_Henry1.doc).
- Verginadis, G. and G. Mentzas, 2004. A light mode framework for e-Government service workflows. Electronic Government, 1: 420-438.