A GATEWAY-CENTERED WORKFLOW ROLLBACK DECISION MODEL TOWARD AUTONOMOUS WORKFLOW PROCESS RECOVERY

In enacting a workflow process model, it is very important to control and trace each instance's execution as well as to keep it recoverable. Especially, the recoverability issue implies that the underlying workflow management system is able to not only provide the automatic error-detection functionality on its running exceptions but also to equip various autonomous recovery mechanisms to deal with the detected exceptional and risky situations. As a theoretical approach to resolve the autonomous workflow recovery issue, this paper tries to formalize a rollback-point decision tree structure based upon gateway-activities of a corresponding workflow process model, which is named as a gateway-centered workflow rollback decision model. We strongly believe that the proposed model ought to be one of those impeccable trials and pioneering contributions to improve and advance the capability of recovery in enacting workflow process models.

**Keywords:** information control net; workflow model; control dependency; recoverability; autonomous workflow process recovery; rollback-points sequence

1. INTRODUCTION

In this paper\(^1\), we focus on the safeness (or recoverability) issue (Ma, J. et al., 2011) (Kim, K. et al., 2007) (Park, M. et al., 2015) (Grefen, P., 2002) (Grefen et al., 2006) in enacting workflow process models through a workflow management system. To guarantee safeness on workflow enactment services implies to keep consistency between the virtual status of a workflow instance on the runtime system and the physical status of the corresponding process on the real business world. For the sake of the safeness, it is very important for the system to be supported by autonomous error-detections and self-recovery mechanisms for resolving the unexpected exceptional and risky situations (Ghadge et al., 2013) (Brzeziński et al., 2012). A little more specifically speaking, the paper conceives a novel concept of rollback-points ancestries to be used for the workflow management system to automatically recover and resume the error-involved workflow instances from the exceptional and risky situations. The proposed concept is a theoretical approach to be reified as an autonomous recovery mechanism that is able to support determining a proper point of rollback operations. The rollback operation needs to point to a specific activity (either gateway-activity or task-activity) out of the previous committed activities, and starts rollback up to the determined activity in a corresponding workflow instance confronting the inconsistency situations. The core part of the theoretical approach is a rollback decision tree structure that produces a set of possible rollback-point sequences from a workflow process model, and in this paper we name it ‘workflow rollback decision model’.

In principle, there are two types of rollback decision models, such as gateway-centered rollback decision model and task-centered rollback decision model. A rollback decision model basically specifies a series of rollback sequences as workflow instance recovery information, which can be used for implementing an autonomous self-recovery mechanism to resume the error-involved workflow instances from the exceptional and risky situations. Note that a workflow process model describes a temporal precedence (control flow) of gateway-activities and task-activities, and its related events, such as initiating, terminating, timing, and so on. So, the rollback-point sequences of a workflow process model can be classified into a gateway-centered type\(^2\). In this paper, we try to extensively refurbish the gateway-centered type as the name of the gateway-centered workflow rollback decision model\(^2\) or a task-centered sequence type. That is, once the system detects an error situation, and then it has to make a decision for choosing a rollback-point where the system applies a series of forward-rollback operations to recover the underlying workflow instance. In general, we can use three types of gateway activities (Ellis, C. A., and Nutt, G. J., 1993), such as alternative-gateway, conjunctive-gateway, and iterative-gateway, in defining a workflow process model, and so the gateway-centered workflow rollback decision model proposed in the paper specifies a temporal sequence of gateway-activities that can be helped for the system to choose a reasonable rollback-point, efficiently as well as autonomously. In this paper, we expatiate on the detailed formalism of the gateway-centered workflow rollback decision model with an operational example.

In terms of making up the paper, the next section gives the conceptual background and the scope as well by mainly focusing on the workflow recovery functionality and its related rollback techniques. In the next two consecutive sections, we describe the details of the proposed concept formalizing the gateway-centered workflow rollback decision model, and its

---

1 This paper is fully extended from the conference paper (Park, M. et al., 2015) published on the proceedings of AP-BPM 2015, the Asia-Pacific conference on Business Process Management held in Busan, South Korea.

2 Note that we firstly formalized a gateway-centered type as the name of the gateway-centered workflow rollback-points ancestry model published in the conference proceedings (Park, M. et al., 2015).
related algorithm with an operational example. Finally, we give a summary with a brief description of its related works and conclusions including future works.

2. CONCEPTUAL BACKGROUND AND SCOPE

We know that the self-recoverable capability of a workflow management system depends on its rollback mechanism to be applied to those running workflow process instances. The rollback mechanism used to determine a rollback-point by indicating an activity out of the previously completed activities on a corresponding workflow process instance. However, because the system does not know from where the detected exceptions come, it has to fulfill the rollback operations such as redoing or undoing for all the completed activities to recover the wrong-running workflow process instance. At this moment, we also know that the system ought to be much more efficient if it is able to determine an exact rollback-point by choosing a proper activity as the start point of undoing (or redoing), rather than undoing all the completed activities, which means aborting the running workflow process instance. The following are the possible policies, which are intuitively applicable to resuming wrong-running workflow process instances:

- No rollback-points: kill-and-restart (abort) the wrong-running workflow process instance, which means undoing (or redoing) all the completed activities;
- Rollback-points that are randomly selected out of the completed activities;
- Rollback-points that are selected via the most-recently-completed-first policy;
- Predefined rollback-points that are specified at the build-time (the modeling time) of a workflow process model;
- The ideal policy: rollback-points that are autonomously selected via a rollback-point decision tree structure.

In this paper, we try to dig out one of the ideal policies. We are able to develop a self-recoverable rollback mechanism, the conceptual background of which is based upon the rollback decision model proposed in this paper, in order to eventually realize our goal that is keeping the integrity on workflow enactments. As an ideal policy, we propose a novel concept of workflow rollback sequences, which is based on a workflow risk dependency model firstly defined in (Chun, M. et al., 2011). There are three types of workflow rollback sequences: gateway-centered, task-centered, and hybrid. As the scope of the paper, we focus on the gateway-centered workflow rollback sequence generation approach. In the gateway-centered approach, we reasonably assign a single gateway-centered rollback-point to every activity on a workflow process model, and name it as risk dependency point. The risk dependency point of each activity becomes a rollback-point of the activity when it encounters with any type of exceptional situations. Note that the syntactical structure of a workflow process model is determined by the basic gateway primitives, such as disjunctive, conjunctive, and iterative gateway-activity types, and so the three types of risk dependency points, such as alternative-risk, conjunctive-risk, and iterative-risk dependencies, can be deduced from a workflow process model, and they become the theoretical bases of a gateway-centered rollback-point ancestry model proposed in the paper.

3. FORMALISM FOR THE GATEWAY-CENTERED ROLLBACK DECISION MODEL

We are proposing an autonomous rollback-points selection approach as one of the ideal policies, which is formalized by the gateway-centered workflow rollback decision model. As stated in the previous section, it is possible to deduce three types of workflow rollback decision models by revising the algorithm (Chun, M et al., 2011). Before getting those workflow rollback decision models, it has to algorithmically generate a workflow risk dependency net from an information control net (Ellis, C. A., 1979) (Ellis, C. A., 1983) of the corresponding workflow model. From the generated risk dependency net, it is possible to build those three types of workflow rollback decision models, such as gateway-centered rollback decision model, task-centered rollback decision model, and hybrid. Our focus pays on the gateway-centered rollback decision model, in particular, that produces a series of gateway-centered rollback-point sequences.

3.1 Information Control Net and Its Risk Dependency Model

In order to automatically generate a gateway-centered rollback decision model from a workflow process model, we would use the concept of workflow risk dependencies (Chun, M et al., 2011) that the authors' research group developed and presented in (Chun, M. et al., 2011). The concept of risk dependencies can be defined graphically and formally by generating the workflow risk dependency net and workflow risk dependency model, respectively, from an information control net of workflow process model. The concept of gateway-centered rollback decision net can be deduced from a workflow risk dependency net of the corresponding information control net.

A workflow risk dependency net is for modeling the risky effects on control transition types, such as sequential, conditional or-split and or-join), parallel (and-split and and-join), and loop (loop-split and loop-join) control transitions. In
A Gateway-Centered Workflow Rollback Decision Model

In particular, the three types of gateway-centered control transitions (conditional, parallel, and loop) become important leverages to form three types of risk dependencies in a workflow risk dependency model by revising the risk dependency generation algorithm (Chun, M. et al., 2011). Assume that we have to use a series of special notations, operations and their meanings, which are closely related with the risk dependency analysis and precisely described in (Chun, M et al., 2011). In terms of revising the algorithm, it is necessary to extend the domination-relationship operations (Podgurski, A., and Clarke, L. A., 1990) (Chun, M. et al., 2011) so as to incorporate the concept of gateway-centered risk dependency types, such as alternative-gateway, conjunctive-gateway, and iterative-gateway dependency types. However, we won’t describe any further in this paper, because of the page limitation. In this paper we focus on formalizing the gateway-centered rollback decision model, in particular.

![Figure 1. An Information Control Net and Its Workflow Risk Dependency](image)

By using the primitive operations (walk, dominance, and dependency) (Podgurski, A., and Clarke, L. A., 1990) (Chun, M. et al., 2011) defined in the previous research, it is possible to extract a set of risk dependency relationships from a workflow process model by the risk dependency analysis algorithm (Chun, M. et al., 2011) to be described in the next subsection. Also, the extracted risk dependency relationships (knowledge) (Park, M., and Kim, K. P., 2008) can be formally described through the formal representation of the workflow risk dependency net defined in [Definition 1].

**Definition 1. Workflow Risk Dependency Net** of a workflow process model. A workflow risk dependency net is formally defined as $\Omega = (\varphi, \xi, I, O)$ over a set $A$ of activities and a set $T$ of transition conditions, where

- $\varphi = \varphi_{rd} \cup \varphi_{re}$
  where $\varphi_{rd}: A \rightarrow \wp(A)$ is a multi-valued mapping of an activity to a set of activities, which is strongly, alternately, or multiply risk-dependent on the activity, and implies that the risk of the activity may depend on the mapped activity(s), and $\varphi_{re}: A \rightarrow \wp(A)$ is a single-valued mapping of an activity to the activity that is strongly, alternately or multiply risk-dependent, which implies that the risk of the activity may effect to the mapped activity;

- $\xi = \xi_{rd} \cup \xi_{re}$
  where $\xi_{rd}$: a set of transition conditions, $\tau \in T$, on each arc, $(\alpha, \varphi_{rd}(\alpha))$; and $\xi_{re}$: a set of transition conditions, $\tau \in T$, on each arc, $(\varphi_{re}(\alpha), \alpha)$, where $\alpha \in A$;

- $I$ is a finite set of initial input repositories;

- $O$ is a finite set of final output repositories;

The graphical representation of the workflow risk dependency net is the shape of the directed tree structure. In mapping the workflow risk dependency net diagram into its formal definition, every node in the diagram correspond to each of the activities, including the start and terminate events in its workflow process model, where a solid arrow implies a risk
dependency relationship between two associated activities, which implies that the destination node of the arrow is risk-dependent on its source node. Additionally, each transition condition is positioned on its associated solid arrows. Of course, the workflow risk dependency net is able to accommodate compound activities such as subprocess. Next, we need to determine how to build a workflow risk dependency net from a workflow process model; we developed an analysis algorithm for this by using the special operations such as walk, dominance, and risk dependency (Podgurski, A., and Clarke, L. A., 1990). We won’t describe the details of the conceived algorithm in this paper, because it is out of the scope of this paper. As an example, the right-hand side of Figure 1 shows a workflow risk dependency net in a graphical representation, which is generated from the simple workflow process model of the left-hand side of Figure 1.

3.2 Gateway-Centered Rollback Decision Model

As described in the previous, the novel concept of workflow rollback-points sequences can be realized in a gateway-centered rollback decision net that can be automatically generated from a workflow risk dependency net of an information control net of workflow process model. The gateway-centered rollback decision net is graphically representing a set of gateway-activities and their risk dependency edges. In this section, we formalize the formal definition of gateway-centered rollback decision model via the gateway-centered workflow rollback decision net, $M^g = (\chi^g, \theta^g, I, O)$, as shown in [Definition 2].

**Definition 2. Gateway-Centered Workflow Rollback Decision Net** from a workflow risk dependency net. Let $M^g$ be a gateway-centered workflow rollback decision net, that is formally defined as $M^g = (\chi^g, \theta^g, I, O)$, over a set of gateway-activities, $A^g$, and a set of transition-conditions, $T^g$, where

- $\chi^g = \chi_{de}^g \cup \chi_{an}^g$
  where, $\chi_{de}^g$: $A^g \rightarrow \theta^g(A^g)$ is a multi-valued mapping of a gateway-activity to an another set of gateway-activities, each member of which is a direct descendant having disjunctive, conjunctive, and iterative gateway-centered dependencies, and $\chi_{an}^g$: $A^g \rightarrow \theta^g(A^g)$ is a single-valued mapping of a gateway-activity to an another gateway-activity that is a direct ancestor having disjunctive, conjunctive, and iterative gateway-centered effects.
- $\theta^g = \theta_{de}^g \cup \theta_{an}^g$
  where, $\theta_{de}^g$: a set of control-transition conditions, $\tau \in T^g$, on each arc, $(\alpha, \theta_{de}^g(\alpha))$; and $\theta_{an}^g$: a set of control-transition conditions, $\tau \in T^g$, on each arc, $(\theta_{an}^g(\alpha), \alpha)$, where $\alpha \in A$;
- $I$ is a finite set of initial input repositories;
- $O$ is a finite set of final output repositories;

In order to systematically construct a gateway-centered workflow rollback decision net from a workflow risk dependency net, the authors’ research group has devised an algorithm that is named as the “gateway-centered workflow rollback decision net generation algorithm”. The algorithm is able to produce a gateway-centered rollback decision net, such as “A Gateway-Centered Workflow Rollback Decision Model ($M^g = (\chi^g, \theta^g, I, O)$)”, from a workflow risk dependency net, “A Workflow Risk Dependency Net ($\Omega = (\varphi, \xi, I, O)$)”, described in (Chun, M. et al., 2011). The following is the algorithm that is able to generate a gateway-centered rollback decision net from a workflow risk dependency net:

```
Input A Workflow Risk Dependency Net, $\Omega = (\varphi, \xi, I, O)$;
Output A Gateway-Oriented Rollback Decision Net, $M^g = (\chi^g, \theta^g, I, O)$;
Initialize Global $T \leftarrow \{\emptyset\}$; /* $T$ is a set of traversed nodes. */
    Local $\omega, u \leftarrow \{\alpha_i\}$;
PROCEDURE ROLLBACK-POINTS (In $s$) /* Recursive Procedure */
BEGIN
    $O \leftarrow \varphi_{rd}(s)$;
    IF ($O$ is an empty-set?) Then do EXIT(); end; END IF
    IF ($s = \alpha_i$)
        Then do $\omega \leftarrow s$; $T \leftarrow T \cup \{\omega\};$ end;
    Else do $\omega \leftarrow \varphi_{re}(s)$; $T \leftarrow T \cup \{\omega\};$ end;
    END IF
    FOR ($\forall u \in O$) DO
        SWITCH (What is the type of $u$?) DO
```
Case ‘elementary task-activity’:

Case ‘compound task-activity’:

\[ \begin{align*}
T & \leftarrow T \cup \{u\}; \\
& \text{Call PROCEDURE ROLLBACK-POINTS (In } u \text{);} \\
& \text{break;}
\end{align*} \]

Case ‘split-gateway- activity’:

\[ \begin{align*}
& \chi^g_{de}(\omega) \leftarrow \chi^g_{de}(\omega) \cup u; \\
& \chi^g_{an}(u) \leftarrow \chi^g_{an}(u) \cup \omega; \\
& \vartheta^g_{de}(\omega) \leftarrow \xi_{rde}(\omega); \\
& \vartheta^g_{an}(u) \leftarrow \xi_{rea}(u); \\
& T \leftarrow T \cup \{u\}; \\
& \text{Call PROCEDURE ROLLBACK-POINTS (In } u \text{);} \\
& \text{break;}
\end{align*} \]

Default:

\[ \begin{align*}
T & \leftarrow T \cup \{u\}; \\
& \text{break;}
\end{align*} \]

END SWITCH

END FOR

IF \((T^g \text{ contains all the activities in } \mathcal{A})\)

Then do

RETURN The Gateway-Oriented Rollback Decision Net, \(M^g = (\chi^g, \vartheta^g, I, O)\);

end:

END IF

END PROCEDURE

The Gateway-Centered Rollback Decision Net

Figure 2. The Gateway-Centered Workflow Rollback Decision Net from Figure 1

Table 1. Formal Representation of the Gateway-Centered Rollback Decision Model for the Information Control Net of Figure 1

\[ \begin{align*}
M^g = (\chi^g, \vartheta^g, I, O) \over A, T \text{ /}\# \text{ The Gateway-Centered Rollback Decision Model} \\
A^g = \{a_1, a_3, g^D_p, g^D_s, g^A_p, g^A_s \} \text{ /}\# \text{ Gateway Activities} \\
T = \{d(\text{default}), tc_1, tc_2 \} \text{ /}\# \text{ Transition Condition} \\
I = \emptyset \text{ /}\# \text{ Initial Input Repositories} \\
O = \emptyset \text{ /}\# \text{ Initial Output Repositories}
\end{align*} \]
A Gateway-Centered Workflow Rollback Decision Model

\[ \chi^g = \chi^g_{de} \cup \chi^g_{an} \]
\[ \chi^g_{de}(a_i) = \{g^s_i\}; \quad \chi^g_{an}(a_i) = \{\emptyset\}; \]
\[ \chi^g_{de}(g^s) = \{\{g^s\}\}; \quad \chi^g_{an}(g^s) = \{\{a_i\}\}; \]
\[ \chi^g_{de}(g^p) = \{\{g^p_i, g^p\}\}; \quad \chi^g_{an}(g^p) = \{\{g^p\}\}; \]
\[ \chi^g_{de}(g^a) = \{\{a_F\}\}; \quad \chi^g_{an}(g^a) = \{\{g^a\}\}; \]

\[ \varrho^g = \varrho^g_{rd} \cup \varrho^g_{re} \]
\[ \varrho^g_{de}(a_i) = \{d\}; \quad \varrho^g_{an}(a_i) = \{\emptyset\}; \]
\[ \varrho^g_{de}(g^s) = \{\{tc_1, tc_2\}\}; \quad \varrho^g_{an}(g^s) = \{d\}; \]
\[ \varrho^g_{de}(g^p) = \{d\}; \quad \varrho^g_{an}(g^p) = \{d\}; \]
\[ \varrho^g_{de}(g^a) = \{\emptyset\}; \quad \varrho^g_{an}(g^a) = \{d\}; \]

Figure 2 depicts the graphical representation of the gateway-centered rollback decision net formed in a decision tree with respect to the rollback-points sequences (or ancestries), and it can be used in developing an autonomous rollback-point selection mechanism. Also Table 1 is the formal representation of Figure 2. Especially, \( \chi^g = \chi^g_{de} \cup \chi^g_{an} \) is formally representing the ancestral relationships with risky dependency and effect properties, where \( \chi^g \) is a kind of gateway-activities. These gateway-centered ancestral relationships are eventually used for choosing gateway-centered rollback-points of workflow instances spawned from a corresponding workflow process model.

In summary, we believe that the gateway-centered workflow rollback decision model plays a very important role as a risk recovery mechanism to realize the eventual goal of sustainable and agile workflow process enactments. From the ancestral relationships' information of \( \chi^g = \chi^g_{de} \cup \chi^g_{an} \), we are able to build a series of autonomous rollback-point sequences for each of the workflow instances spawned from a workflow process model. In particular, \( \chi^g_{an} \) possesses the hierarchy of gateway-centered ancestors that dominates every activity in a corresponding workflow process model. Therefore, we can automatically generate a series of rollback-point sequences of each workflow instance so as to be applied to recover from the risky situation where it is faced with. As an operational example, the following lists a series of rollback-point sequences that can be obtained from the gateway-centered ancestral relationships (descendants and ancestors) information, \( \chi^g_{de} \) and \( \chi^g_{an} \), respectively.

- Gateway-Centered Rollback-Point Sequence \( 0 = g^f_a \rightarrow g^s \rightarrow g^s_a \rightarrow a_i \)
- Gateway-Centered Rollback-Point Sequence \( 1 = g^f_p \rightarrow g^p \rightarrow g^p_a \rightarrow a_i \)

4. RELATED WORKS

To support the agility in workflow process operations and enactments, it is important for the system to provide the essential functionalities chosen from the agility concept defined in (Taylor and Francis, 2014): detection (i.e., the ability to diagnose the divergence between the real situation and the expected situation), adaptation (i.e., the ability to define and implement a new and dedicated behavior coherent with the diagnostic), reactivity (i.e., the ability to ensure detection and adaptation fast enough to ensure the achievement), and effectiveness (i.e., the ability to ensure detection and adaptation in a pertinent way and with precision). For the sake of realizing and improving these essential functionalities, several research trials are mentioned in the literature, including our approach to resolve the agility and sustainability issues with respect to detecting the information flows (Rosenkranz, C., and Holten, R., 2011), adaptive and reactive risk managements (Chun, M et al., 2011) (Brzeziński et al., 2012), and the quality of effective services (Park, M et al., 2015) in workflow process operations and enactments.

Furthermore, in order to improve the detecting information flows in the workflow process enactments, we need to adopt the methodology developed by C. Rosenkranz and R. Holten (Rosenkranz, C., and Holten, R., 2011), in which analysis and design of information flows are performed in organizations in a structured way. In the workflow process-driven organization, the efficiency of a value chain is strongly influenced by the accurate set up of information flows on workflow processes. To make the workflow processes more effective and efficient, one needs to understand the information flows that are currently available and how information flows should be designed for a given workflow process. As a future research work, our approach needs to integrate with the methodology as a means of detecting the inconsistencies in information flows between the activities in a workflow process model.

Moreover, our approach (Park, M. et al., 2015) including this paper has hardly investigated the risk situations and classifications on the workflow processes management perspective, and it simply describes the risk situations as the inconsistencies (Brzeziński et al., 2012) (Ma, J. et al., 2011) between the virtual-world manifested and the real-world
manifested on workflow process enactments. In the next step of our approach, we need to actively adopt the risk analysis and management methodologies of workflow and business process management systems, which have been fulfilled by the workflow risk management arena. In other words, we need to explore the more precisely detectable inconsistencies and the more delicate risk situations that can be detected by the RFID-enabled activities in a corresponding workflow process model. By conceptually connecting their outcomes to our approach, we can expect to develop a holistic, systematic, and quantitative risk assessment process for measuring the overall risk behaviors on workflow process enactment and management activities.

5. CONCLUSIONS

So far, this paper has deployed the concept of gateway-centered workflow rollback decision model and its usage for autonomous workflow recovery mechanisms to achieve the sustainable and agile workflow process enactment services. Based upon the theoretical concept, we are able to not only generate a series of gateway-centered workflow rollback-point sequences for a workflow process model, but also extract the gateway-centered rollback-points ancestral relationships to be used for the autonomous workflow recovery mechanisms. As stated previously, in resolving the recoverability issue of workflow process enactment services, it is very important for the system to provide autonomous error-detection and recovery functionalities on its running exceptions as well as very safe self-recovery mechanisms on the exceptional and risky situations. Therefore, we are certainly sure that gateway-centered rollback-point sequences from a gateway-centered rollback decision model proposed in the paper can produce valuable workflow risk dependency knowledge that will be eventually used for realizing the sustainable and agile workflow processes enactment services.

ACKNOWLEDGEMENT

This research was supported by the Basic Research Funds of the National Research Foundations, ID. 2013-0625, Republic of Korea. Also, it was partially supported by Kyonggi University’s Graduate Research Assistantship 2016.

REFERENCES


