Research on The Relationship Between Workflow net based on Petri nets and UML.

Wang Jingguang  
The central university of finance and economical Beijing China  
wangjg@cufe.edu.cn

Abstract

In this paper, We have studed the relationship of key elements between workflow net based on petri net and UML by the structural property, and set up the relationship of one corresponded with one. The aiming at enrich the deficiency of UML in modeling course on MIS with the characteristic of workflow net which can describe the different process of business.

Keywords: workflow net, UML, relationship, petri

1. Introduction

UML as the object-oriented Modeling tool have already been applied in the process of modeling and analysis of developing on the Enterprise Information System. However, it is hard to reach the consistent results perfectly when we try to descript a whole business process of enterprise because of the respectively focuses of the different diagrams in UML. Workflow Nets based on Petri Nets can model for various granularities of business processes in details. If we can use modeling with Workflow Nets to extend UML, they will provide the strong support for system modeling and analysis in the process of developing the Enterprise Information System with UML.

2. UML and the essential elements of workflow Nets

The transition in the workflow nets may be regarded as the operation on the objects and can be used to descript the dynamic characteristics of an objects. It corresponds to the class operation in UML (Figure 1). A transition usually to be defined an executable statement or procedure for some services.

![Diagram](image.png)

Figure 1 transition and operation

The place in Workflow nets can be regarded as a state of an object or represents pre-condition or post-condition when an operation (transition) is executing. It corresponds to the class’s state In UML (Figure 2). The state refers to the condition and status during the object’s living time and the object should match certain of conditions during this time, executing some activities or waiting for some events.
The directed arc in the workflow nets represents the relationship between two states. The object executes specified action in the first state and goes into the second state when an event happens or a specified condition is matched. The transition in UML is relationship between two states, which shows the path from an action state to the other. The directed arc and combination of transition in the workflow nets is corresponding with transition in the state chart of UML.

![Diagram](image)

**Figure 2** model of workflow net and state chart in UML

The token in workflow nets represents an instance of a process state, it describes explicitly the process state by the number of tokens in the Place. The distribution of tokens describes the specified state and its change shows the systematic evolving process. There is no element in UML state chart correspond to token, so it is only be shown implicitly or by other methods when the systematic evolving process need be described.

### 3. UML and the triggering mechanism of workflow nets

By the triggering mechanism of workflow nets, the auto-triggering is referred to that activities will be triggered when they are enabled, and it corresponds to The nontrigger transition in State-chart of UML. The nontrigger transition is also called completion transition which refers to that a action will the execute exit motion when its source state is over, and then the next action will be triggered implicitly (automatically). The auto-trigger in workflow nets and the nontrigger transition in State-chart of UML show the auto-state will begin to execute without is triggered by the outside conditions and not give the special triggering time only when it is enabled.

Manual triggering, message triggering and timed triggering in workflow nets represent that triggering is no the independent behavior. When an activity is enabled, it will be triggered only when the external event associated with it happens. Manual triggering is activated by the system operator selects the appropriate working item. Message triggering is launched by message from outside of system. Timed triggering refers to that the action is activated by the time unit which controlling time when it is enabled. All above mentioned triggers are corresponded to the event-triggered state-chart in UML (Figure3). The time event “after (2 seconds)” is corresponding to the timed triggering in workflow nets. The event-triggered “Collision [is Thread]” with the guard conditions is corresponding to “event-triggered”. Meanwhile, the nontrigger transition without the guard conditions is corresponding to “auto-triggered” in workflow nets.

In UML, an event is a product of stimulation which can trigger the state transition. This event is either synchronous or asynchronous. Event in UML includes signal, call, time process and a change of state. The signal and call may take parameters whose value is set by transiting the express including the guard conditions and actions. Therefore, UML may provide more abundant semantics than that of workflow Nets in the aspect of nontrigger transition. For
example, it can provide a guard condition to evaluate triggering.

![Figure 3](image)

In UML, the guard condition is a Boolean expression with square brackets following the triggering event which is evaluated after the triggering event activating transition happens. However, if the transition is reactivating, the guard condition will be evaluated again. Generally, as long as these conditions don’t overlap, it is likely to have several transitions derived from the same source state and caused by the same event. Through the guard condition is evaluated only one time when its transition is triggered, in fact the changeable events are evaluated implicitly and constantly. Thus it can be seen that the nontrigger transition with the guard condition in UML have the more clarified format than that of manual triggering, message triggering, timed triggering in workflow nets, so it is easier to understand the condition provided by the triggerless transition. Even if the persons is not familiar with the specified contents of business process, they can have a clear idea of the specified time of transition and the way how message should match with event to execute the transition.

4. UML and the essential components of the workflows

4.1 UML and The serial components

The essential component in workflow nets may be used to define a serials of the activities with the unchangeable sequence, and it is composed of a sequential access. It’s similar to create modeling for the monotone, sequential controlling activities in state-chart of UML (see the figure 4)
4.2 UML and The concurrent components

The concurrent components are used to define the activity branch with the unfixed sequence and the synchronous and asynchronous execution. To describe such kind of the parallel branch activity, the workflow nets are based on the place and transition and utilize the workflow primitive that is “And-brancht” and “And-joint”. Compared to UML, “And-branch” in Workflow Nets is corresponding to “concurrent fork ” in UML, and “And-joint” to “concurrent join”.

In UML, the horizontal (or vertical) line called the synchronous club is used to represent the branch and joint of the concurrent control flow. The concurrent branch is composed of an entry transition, a synchronous club and two (or more than one) exit transitions. The concurrent join is composed of two entry transitions, a synchronous club, and an exit transition. The corresponding relationship between the concurrent components in the workflow Nets and the “concurrent fork” and “concurrent join” in UML is showed in Figure 5.

![Figure 5](image_url)

The “And-joint” and “And-branch” consisting of the concurrent components are the same as the “concurrent fork” and “concurrent join” in UML. Joint and branch is balanceable. The number of flows describing to exit fork should be the same as that describing to entry the corresponding joint exactly.

4.3 UML and the Selection components

The selection components are used to define the branch activities which constraint and disjoint each other. Generally, this kind of activity be chosecd one or more from more choice according to the practical instances of execution. The “Or-branch” and “Or-joint” are utilized to describe the selection components. The “Or-branch” consists of “the implicit Or-branch” and “the explicit Or-branch”. Therefore, the selection components consist of that of “the explicit Or-branch” and “the implicit Or-branch”.

2054
The diamond is used to represent branch of the activity Diagram in UML. A branch may have an entry transition and more than one exit transitions. A Boolean expression (the guard expression) is taken with each exit transition. The branch in the activity Diagram isn’t divided into “implicit” and “explicit” unlike the workflow Nets. They are distinguished from each other by judging whether the Boolean expression is on the exit transition or not. The “explicit expression or fork” selection components are corresponding to the branch with the Boolean expression of the activity Diagram in UML (see the figure 6).

The other diagram in UML has also definition representing condition. For example, usually uses the Boolean expression to judge whether a message can be executed or not in the interaction diagram. When model for condition, in the collaboration diagram adds a condition substatement before the sequence number. For example, \([x>0]\) is used to judge whether the condition is true or false. The message is only sent if the condition is true. A branch’s choice path takes the same sequence number. But each path should be distinguishable uniquely by the unoverlapping condition. The sequence diagram doesn’t show not only link (or chain) among objects but also the sequence number of a message explicitly. Its sequence number hides the physical sequence from top to bottom in diagram. A choice branch path of sequence diagram is represented by the several different messages from one point. Comparatively, branch described by the collaboration diagram is more complicated than that of the sequence diagram.

There is no the corresponding symbol with “Or-joint” of the workflow in UML. It is only represented by the symbol “I” in Figure 6. However, “I” has the corresponding relationship with “Or-joint”. Through UML has no the explicit symbol to “Or-joint”, we can know “I” is “Or-joint” from the figure obviously. Actually, the workflow primitive also utilizes it to represent that two (or more than one) control flows flow into one place. It is different between them that it refers this kind of state as “Or-split”. However, they are same in nature.

4.4 UML and Recurrence components
The recurrence components in workflow Nets is used to define activity which is executed several times. It utilizes a “explicit OR branch ” primitive, whereas it is used frequently, in order to achieve the iteration (recurrence), that the value of iterative unit is set by an action
state and revised by the another, meanwhile, a branch is used to judge if the iteration is over or not. The corresponding relationship between the recurrence components and the activity diagram in UML is shown in Figure 7.

![Figure 7](image_url)

The recurrence in workflow Nets is most similar to the iteration in the activity diagram of UML. The former utilizes "explicit OR branch", whereas the latter utilizes "branch". The former judges if recurrence continues to execute according to that token moves either C2 or C4, whereas the iteration in activity diagram of UML is judged if do that according to the guard expression.

The interaction diagram may also create model for the control flow including iteration and branch In UML. The iteration, which is usually implement by adding an iteration expression before a sequence diagram, in interaction diagram of UML represents an iterative sequence of message. For example, \([i:=1\ldots n]\) (for collaboration diagram, it is represented by adding the sequence number before an iteration expression). And then, this message will be iterated according to this given expression. If you don’t want to account for the detail of iteration, you can add "\(*\)" before message.

5. Conclusion

The initial objective and elementariness are different in workflow nets and UML, and There is many differences when they are used to modeling. But we have found by research that there are many similarities and even same things at the aspects of basic elements and internal mechanism between them. If they can joint each other and learn from others' strong points to offset one's weakness, it is evident to achieve self-optimization for each other. For example, now that the activity diagram of UML is used to reflect control flow from activity to activity, it is feasible to reflect the evaluative process dynamically by adding such elements as token of the workflow nets in the activity diagram. In addition, in order to improve the adaptability that the workflow system is to the enterprise application, the nontrigger transition with expression of guard condition is adopted to describe the specific condition of manual triggering, message triggering, timed triggering in detail.
References

