The Workflow-Based Modelling Method for Spacecraft Automatic Testing Process

YU Dan\textsuperscript{1}, LI Zhong-wen\textsuperscript{2}, YE Gang\textsuperscript{3} and MA Shi-long\textsuperscript{4}

\textsuperscript{1,2,3,4} National Lab of Software Development Environment, Beihang University, Beijing, P.R.China.

Abstract - Nowadays workflow technology becomes more and more popular, which simplifies the application development and maintenance work, therefore enhancing efficiency. The process of spacecraft testing is very complex, and still depends on manual work to a large extend, so this paper introduces workflow theory into spacecraft testing area which brings forward a new opening architecture and helps to resolve many problems during the whole testing process. In addition we model the spacecraft testing process and present a testing workflow description language.

Keywords: Spacecraft testing, Process modelling, Workflow, Conditional directed graph, XML

1 Introduction

The workflow originates from the field of production and office automation, whose goal is to divide the work into well-defined tasks and roles, and to implement and monitor these tasks according to certain rules and processes, thus to increase efficiency, reduce production costs, advance the level of operation and management, and strengthen the competitiveness of enterprises. Workflow Management Coalition presented the workflow and its modelling definition: workflow is the automation about all or partial business process, and the file, information or task are circulated in the light of certain process rule to realize the coordination among the members so that the whole goal of the business realizes, and the workflow modelling is to abstract the reality business process and use a formalized and computer-handled way to represent, as this formal result is called as workflow model\textsuperscript{[1,2]}.

As the same with any engineering projects, the spacecraft in the development stage can not be separated from a large number of trials from the beginning to the end. In order to recognize whether the spacecraft performance and functionality can meet the design requirements, various indicators of spacecraft are needed to test. Here the spacecraft testing refers to the use of various software and hardware systems to test each computer subsystem in the spacecraft, to detect if each spacecraft internal subsystem and between the subsystems work in right way or not.

Spacecraft testing has the following characteristics: (1) Testing business process is complicated and changeable. (2) Testing requires high real-time response. (3) Each business process of testing is similar. For the moment spacecraft testing still depends on manual work to a large extend, thus resulting in the low efficiency considering the above testing characters\textsuperscript{[3,4]}. So workflow technology, whose main function is to realize the business process automation, can help to solve this problem.

Presently the workflow technology becomes more and more important; however the study of workflow on spacecraft testing is not too common. This paper introduces the workflow technology into spacecraft testing, which can greatly improve the automation level of spacecraft testing, and meet the frequently changing requirements.

The rest of the paper is organized as follows. Section 2 introduces spacecraft testing workflow. Section 3 presents the architecture of spacecraft testing workflow. Section 4 and section 5 propose test process modeling and a testing workflow description language respectively. Section 6 gives the concluding remarks.

2 Spacecraft Testing Workflow

Workflow products can be divided into the following four types according to the purposes and technical characteristics: administrative workflow, ad hoc workflow, collaborative workflow and production workflow\textsuperscript{[5]}. These products can provide good solutions for general applications, but they are not suitable for spacecraft testing because of: (1) Real-time requirement. The spacecraft testing requires high real-time, but obviously general workflow products can not meet this demand. (2) Safety requirement. Safety here means that we need provide adequate mechanisms to ensure that the course of testing on the spacecraft will not cause any damage. Ordinary workflow products can not satisfy this demand. (3) Description of resource equipments and access control. Spacecraft testing process involves numerous physical equipments, so it is necessary to manage these equipments and provide them for automatic process. The existing workflow products take into account external resources, but the problem of how resources provide services for business process has not been resolved.

So it is essential to develop workflow management system specifically for spacecraft testing. In this paper, we first present the architecture of spacecraft testing workflow, then model the testing workflow process with characteristics of spacecraft testing business, and finally give the process description language for spacecraft testing.
3 The Architecture of Spacecraft Testing Workflow

The spacecraft testing process control can be separated from its business when workflow is introduced into spacecraft testing, so we apply hierarchical structure to manage spacecraft testing. The architecture of spacecraft testing workflow is showed as figure 1.

![Figure 1 Architecture of Spacecraft Testing Workflow](image)

The business level includes a business component library, as well as the registration center of various components in the library. Each testing atom component in the component library completes a specific testing operation. Thus a testing business can be achieved only calling to different testing atoms according to some logic combinations. For example, sending the satellite a command, or determining parameter values sent by satellite is correct or not can be defined as testing atomic component. When a new business appears, it is only to define one or a series of testing atoms, and once define a new test atom, it can be reused by all testing processes.

Another important component in the business layer is the registration center. All components must be registered in the center, and it provides the indexes of testing atomic components, additionally gives the calling method of each atom.

The workflow layer is responsible for the introduction and parsing the specific testing process. The layer calls testing atoms in accordance with the corresponding activities according to the process definition, including transmission parameters of the entrance to the testing atomic components, and access the return data of testing atomic components for later process. The workflow services must also provide a group of interfaces, including creation, activation, pause, and termination of the testing workflow instances.

There is a component access agent between the business layer and workflow layer, which is designed to shield the different specific implementation of testing atomic component and access policy, such as the RMI, RPC, etc. Every testing atom has its own access agent on the end of implementing engine, and the interaction mechanism is uniform between the engine and agent. Thus the agent can simplify the design of testing workflow engine on the one hand; on the other hand every testing atom can be realized by different languages.

4 Spacecraft Testing Process Modelling

Spacecraft testing process modelling consists of the spacecraft testing process data definition and spacecraft process control.

4.1 Testing process data definition

Spacecraft testing process data definition consists of two parts: the definition of data types and transition relations of data in the course of spacecraft testing. The data type in the testing process can be string, integer, float, long, byte, and etc. The conditional expressions in implementation of activity transition need to use these standard data types to construct expressions in order to computer under conditions. Moreover there is a special data type, `unknow_type`, which can be one of the above standard type, or the arbitrary type defined by users.

![Figure 2 Data Function Domain](image)

All the data used in the process of spacecraft testing must be defined before use, including the declaration of data type, as well as the identifier in accessing the data. The data can be defined on three levels: business process level, sub-procedure level and activity level, but used in its own scope, which is directly owned by the definition of the entity meta-model. The meta-model definition of each level will be given in details in later chapter. The declared data in the business process level is effective for all sub-procedures, similar to the global variables, the data declared in the sub-procedure is only effective for the activities in this procedure, and data defined in the activity is just effective for this activity.

As shown in figure 2, if the identifier of a data in the business process level is defined as `var`, and procedure `Procedure1` has also defined a data identifier `var`, the keyword of business process will be required, such as BP.var, when the activity in `Procedure1` wants to access the `var` in the
business process. In addition, if at the same time a sub activity Activity1 in Procedure1 has defined a data identifier var, the keyword of business process will be required equally, such as BP.var, when Activity1 wants to access the var in the business process, and the keyword of sub procedure will be required, such as SP.var, when Activity1 wants to access the var in Procedure1.

4.2 Spacecraft Process Control Model

We use the conditional directed graph (CDG) to model the spacecraft testing workflow process[6-7], and firstly give the definitions of CDG.

**Definition 1** Node: The node refers to the activity node in the spacecraft testing workflow.

**Definition 2** Condition transition arc (C-arc) and unconditional transition arc (NC-arc): When the transition between the nodes is conditional, the transition arc between the two nodes is named as conditional transition arc, represented by →C. When the transition between the nodes is unconditional, the transition arc between the two nodes is named as unconditional transition arc, represented by →.

**Definition 3** The CDG of spacecraft testing workflow is denoted as \(G = \langle N, A, C \rangle\), where

1. \(N = \{n_1, n_2, n_3, \ldots\}\) is the node set.
2. \(A = \{ (n_i, n_j, c_k) \}\) is the transition arc set,
   \(A \subseteq N \times N \times C\), \(A = CA \cup NCA\), \(CA \cap NCA = \phi\). CA is C-arc set. NCA is NC-arc set. If \(C_k\) is \(\phi\), the transition arc is NC-arc.
3. \(C\) is the set of conditional expressions, and includes a empty condition \(\phi\).
4. \(G\) has a single start node and a single end node.

After the definition of CDG, we can decompose the graph into the following three models: sequence model, split model and merge model.

- **Sequence Model**

  ![Figure 3 Sequence Model Graph](image)

  Sequence model is the simplest model to understand. It refers to the ordinal implementation according to the scheduled list of nodes. As shown in figure 3, only when the conditions c1 is met, the A→B happen. B→C always has not transferred because the transition condition on B→C is always true.

- **Split Model**

  Split model refers to performing multiple branches in parallel after carrying out a step. As shown in figure 4, there are three branches after executing node B. The transition conditions of each branch will be calculated, and the branch of which the result is false will be abandoned. That is, if the result of c1 is true node C will perform, and the transition condition of B→E is empty, node E will certainly be executed in accordance with the principle that empty is true.

  ![Figure 4 Split model graph](image)

- **Merge Model**

  Merge model refers to converging into a path again after split. The relations between branches have two kinds: AND and XOR.

  1. When the relationship is AND, simultaneous aggregation method is used. Namely, when AND aggregation appears, the engine will wait until the implementation of all precede nodes are finished, and then make the AND calculation of all transition condition after obtaining each transition condition. As showed in figure 5, if node A, B and C are valid nodes, only when node A, B, C has carried out, and the transition conditions c1, c2 and c3 are real, node D will be executed. Any of conditions c1, c2 and c3 is false, node D will be disabled and lead to the failure of the following path.

  2. When the relationship is XOR, there is no need for the synchronization of all precedes nodes. After completion each precede node will calculate its transition condition. If the condition is true, transition happens directly. If the condition is false, transition can not happen. As the above figure, once node B finishes, condition c2 will be calculated. If the result is true, B→D happen; on the contrary, there is no transition.

5 Spacecraft Testing Process Description Language

In order to standardize the spacecraft testing process, we define spacecraft testing workflow description language (TWDL) based on XML in view of scalability and flexibility, and use XPDL as a reference[8]. Figure 6 shows the XML-schema of TWDL. For each elements of TWDL, we give the description segment with XML language.

From this figure, we can see that TestingActivity is the basic structure to construct a testing workflow, which build up
the TestingActivities. In addition, TestingActivities is the element of TestingWorkflowProcess, which composes the TestingBusinessProcesses at last. In the following, we give the detail interpretations of every element.

- **Business Process:<TestingBusinessProcesses>**

  The XML representation of <TestingBusinessProcesses> is as follows:

  ```xml
  <xsd:element name="TestingBussinessProcesses">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="xsd:Header" minOccurs="0"/>
        <xsd:element ref="xsd:DataFields" minOccurs="0"/>
        <xsd:element ref="xsd:TestingWorkflowProcess" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  ```

  The elements in <TestingBusinessProcesses> include:

  - **<Header>:** The head information of testing business, including the preference level, description and etc.
  - **<DataFields>:** The data declaration in business level, including data type and identifier.
  - **<WorkflowProcess>:** A testing business process includes this component (from zero to infinite).

- **Sub-procedure:<TestingWorkflowProcess>**

  The XML representation of <TestingWorkflowProcess> is as follows:

  ```xml
  <xsd:element name="TestingWorkflowProcess">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="xsd:Information" minOccurs="0"/>
        <xsd:element ref="xsd:FormalParameters" minOccurs="0"/>
        <xsd:element ref="xsd:DataFields" minOccurs="0"/>
        <xsd:element name="TransitionConditions">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="StartActivity"/>
              <xsd:element name="FinishActivity"/>
              <xsd:element name="TransitionCondition"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="TestingResources">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="TestingResource" minOccurs="0" maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="Description"/>
                    <xsd:element name="Address"/>
                    <xsd:element name="Port"/>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element ref="xsd:TestingActivites" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  ```

  The elements in <TestingWorkflowProcess> include:

  - **<Information>:** The description information of sub-procedure.
  - **<FormalParameters>:** The input and output parameters of sub-procedure.
  - **<DataFields>:** The data declaration in sub-procedure level, including data type and identifier.
  - **<TransitionConditions>:** The transition information in all activities, including the start activity, end activity and transition expression.
**TestingResources**: Describe the testing resources information in testing process.

**TestingActivities**: Define all activities in this sub-procedure.

- Activity **TestingActivity**

  The XML representation of **TestingActivity** is as follows:

  ```xml
  <xsd:element name="TestingActivity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="xsd:Information" minOccurs="0"/>
        <xsd:element ref="xsd:DataFields" minOccurs="0"/>
        <xsd:element ref="xsd:Implementation" minOccurs="0"/>
        <xsd:element ref="xsd:TransitionRestrictions" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  
  The elements in **TestingActivity** include:

  - **Information**: The description information of activities.
  - **DataFields**: The data declaration in activity level, including data type and identifier.
  - **Implementation**: Define the specific execution of this activity.
  - **Transition Restrictions**: The transition relations in activities, including AND or XOR.

7 References


6 Conclusions

The most prominent characteristic of workflow is the business process automation. This paper introduces the workflow into the field of spacecraft automate testing, and proposes an open and scalable architecture, also presents a workflow description language to model spacecraft testing processes. The workflow technology provides a good solution for the problem of complex and volatile business in the process of spacecraft testing and makes testing atomic components reuse.