ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

Visual Editor for Workflow Rules In Salesforce

Aniket D. Khaire[1], Manojkumar Bhosale[2], Tejbir Singh[3], Prof. S. P. Godse[4]

Student, Computer Department, Sinhgad Academy of Engineering, Pune, India[1,2,3]

Assistant Professor, Computer Department, Sinhgad Academy of Engineering, Pune, India [4]

Abstract: Nowadays interactive systems cannot be developed isolated from workflows and the application domains they are embedded in. Unfortunately, a conceptual gap between design representations for interactive systems and workflow specifications exists. This gap leads to the situation that it can be checked in a very late stage of development whether an interactive system actually fits into a particular workflow or not. If not, redesign is required and the development costs may be increased dramatically. In order to avoid this worst case, in this paper we introduce a framework and environment that allows the refinement of workflows to system functions at an implementation-independent layer. At the heart of this new approach lies a single page interaction model that facilitates rich interactivity. This paper describes how such a visual editor can prove to be useful for quicker generation, and understanding of complex workflows. This paper also gives the various Javascript libraries that would prove useful in developing a better User Interface. A simple design for visual editor is presented in order to demonstrate our approach. Our prototype implementation is discussed at the end of the paper.

Keywords: Salesforce, Apex, Visualforce, Visual Editor, workflow, WSDL ,Metadata API.

I. INTRODUCTION

The term "cloud" originates from the world of telecommunications when providers began using virtual private network (VPN) services for data communications. Cloud computing deals with computation, software, data access and storage services that may not require end-user knowledge of the physical location and the configuration of the system that is delivering the services. Cloud computing is a recent trend in IT that moves computing and data away from desktop and portable PCs into large data centres.

Salesforce.com is a software as a service (SaaS) company that distributes business software on a subscription basis. Salesforce hosts the applications off site. It is best known for its Customer Relationship Management (CRM) products.[6]

The Force.com platform allows external developers to create add-on applications that integrate in to the main Salesforce application and are hosted on salesforce.com's infrastructure.

These applications are built using Apex(a proprietary Java-like programming language for the Force.com Platform) and Visualforce(anXML-like syntax for building user interfaces in HTML, AJAX or Flex).[6] The construction of visual editors is typically addressed in two ways: (1) structure oriented (syntax oriented) visual editors, and (2)general-purpose visual editors. While the first approach considerably restricts the user freedom, the second approach requires visual parsing which still remains a considerable problem'. Ideally, we would prefer hybrid editors that support both structured and free form editing.[1][2]

II. OVERVIEW

The Visual Editor is a visual tool for creating, managing and reviewing process definitions. Straightforward and simple, Visual Editor let users quickly create workflow process definitions, check and store them for further use. Once a process definition is proved valid, it can be imported/referenced into new ones, thus shortening the time and effort needed to define the workflow process. Visual Editor provides several views:

1. for business managers to change the control flow of a process,
2. for technical experts to invoke services, or others.

Visual Editor also makes it easy to visualise the flow, in which, particular actions will get triggered. The process of understanding complex business logic is eased, by the production of Visual Editor. It truly signifies the well known proverb, “A picture is better than thousand words”.

III. SINGLE-PAGE WEB DESIGN

The visual editor to be constructed would be a single webpage application, which is an extreme way to build up a web application in one single physical webpage as the all-in-one page solution with no navigation, just based on already mentioned dynamic and asynchronous techniques. The single page web design limits the structure of the application to fit into only one single physical web page and suppresses the further navigational capabilities. This approach puts the developer before some substantial challenges. It is demanded to design user interface which would show all important information and controls to manipulate data in one single page. Page design has to be elaborated to maximize page decomposition into distinct components encapsulating main use cases. Additionally, most of these richer features cannot be implemented without using additional technologies such as dynamic HTML, JavaScript, Ajax or another. These technologies of course require additional knowledge to be learned.[3][5]

IV. WHAT IS WORKFLOW?

Workflow is a word that means different things to different people. According to [4], workflow is the computerized facilitation or automation of a business process, in whole or part. Workflow is a reliably repeatable pattern of activity enabled by a systematic organization of resources, defined rules and mass, energy and information flows, into a work process that can be documented and learned. Workflows are always designed to achieve processing intents of some sort, such as physical transformation, service provision, or information processing. [4]

Advantages of Workflow

Workflow can enhance the process of interactive software development in the following four areas:

1. Controlling user interface: it controls how user interface behaves and activates or deactivates different part of the UI automatically based on the rules.
2. Business rules: workflow helps managing complex business rule and making it very easy to change and update the rules.
3. Adding features: workflow helps us reduce the time needed for adding new features to the system and also less documentation updating is needed.
4. Monitoring and logging: powerful automatic graphical workflow monitoring tools make it very easy to log and monitor every parts of application without writing any code, so this can save so much development time.

V. WORKFLOW RULE IN SALESFORCE

Each workflow rule in Salesforce consists of:

- **Criteria** that determine when Salesforce executes the workflow rule. Any change that causes a record to match this criteria can trigger the workflow rule—even changes to hidden fields. Criteria can also be termed as conditions.
- **Immediate actions** to take when the workflow rule executes. For example, Salesforce can automatically send an email that notifies the opportunity team when a new high-value opportunity is created. Immediate actions are the actions that are performed as soon as the criteria is met.
- **Time-dependent actions** that Salesforce queues when the workflow rule executes. For example, Salesforce can automatically send an email reminder to the account team if a high-value opportunity is still open ten days before the specified close date.

When Salesforce triggers a workflow rule that has time-dependent actions, you can use the workflow rule to monitor and cancel pending actions.

Workflow automates the following types of actions:

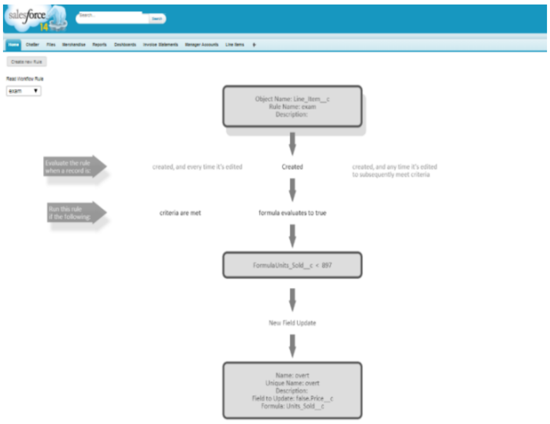
- **Email Alerts**—Send an email to one or more recipients. For example, automatically send sales management an email alert when a sales representative qualifies a large deal.
- **Tasks**—Assign a new task to a user, role, or record owner. For example, automatically assign follow-up tasks to a support representative one week after a case is updated.
- **Field Updates**—Update the value of a field on a record. For example, automatically change the Owner field on a contract three days before it expires.
- **Outbound Messages**—Send a secure configurable API message (in XML format) to a designated listener. For example, automatically initiate the reimbursement process for an approved expense report by triggering an outbound API message to an external HR system.[6]

VI. IMPLEMENTATION

Visual Editor allows developers to create new workflow rules, also read & edit existing rules and their actions, using a single screen application. .[1]

Visual Editor is basically a MVC based tool which has a Visualforce page as front-end and is connected to a controller class written in apex. The model contains all the Custom Objects stored on Developer's Org and also various Static Resources needed. The front end as shown in figure 1 shows all essential elements needed to create a rule. These required elements for a workflow rule can be entered as follows:

- **Object name:** This is the name of the Custom Object on which rule is to be created.
- **Rule Name:** It's a mandatory element to identify and further to read any workflow.
- **Evaluation Criteria:** It determines when to execute the workflow rule. It can be of any one of the following three types: Created, Created and everytime its edited, Created and every time it is edited to subsequently meet the criteria.
- **Rule Criteria :** It is the actual condition (mathematical formula) which yields a Boolean value to determine whether to perform the action or not. It can entered as any one of the following two types: Criteria are met OR Formula evaluates to true.
- **Action :** It is the task to be performed after successful fulfillment of the above mentioned criteria. (see figure 1)



exam

Object Name: Line_Item__c
Rule Name: exam
Description:

created, and every time it's edited
Created
created, and any time it's edited to subsequently meet criteria

formula evaluates to true

FormulaUnits_Sold__c < 897

Name: overt
Unique Name: overt
Description:
Field to Update: false.Price__c
Formula: Units_Sold__c

Fig. 1: Example of a Workflow Rule in Visual Editor

As seen from Figure-1 Visual editor provides us a very easy and interactive interface to work with. It becomes easier to grasp and understand each and every element of the workflow rule. Once you create a rule, you can manage it in Salesforce, edit its attributes, activate, deactivate, delete, or run it at will. Thus Visual Editor, provides a compelling, and a tempting interface to the users, to develop more advanced and complex workflows through it.[1][6][9]

The implementation work is based on two systems:1)Front-End(User-Interface).2)Back-End(Deployement,Business Logic).

7.1 User-Interface
Our prototype implementation of the user-interface to Visual Editor is implemented using Visualforce framework. Visualforce is a user-interface development environment which supports the construction of interactive, graphical interfaces. The main attractions of Visualforce are its: (1) interpretive environment which supports fast prototyping, and (2) graphical constraint system which aids in dealing with layout constraints.[9]

7.2 Back-End
Our prototype implementation of the back-end to Visual Editor is implemented using Apex programming language. After creating the workflow rule in visual editor, Xml file is generated at the back-end which basically consists the code of a workflow rule. That Xml file is then deployed at developers org. The deployment of Xml file is done through various web services like WSDL, SOAP API/ Metadata API. While creating workflow rule in visual editor, code will be automatically generated at the back end.[7][9]

VII. VISUALFORCE

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Force.com platform. The Visualforce framework includes a tag-based markup language, similar to HTML. Developers can use Visualforce to create a Visualforce page definition.[8]

A page definition consists of two primary elements:
1) Visualforce mark-up.
2) A Visualforce controller.

Visualforce mark-up consists of Visualforce tags, HTML, JavaScript, or any other Web-enabled code embedded within a single <apex:page> tag. The mark-up defines the user interface components that should be included on the page, and the way they should appear. A Visualforce controller is a set of instructions that specify what happens when a user interacts with the components specified in associated Visualforce mark-up, such as when a user clicks a button or link. Controllers also provide access to the data that should be displayed in a page, and can modify component behaviour.[7][9]

Developers can use Visualforce pages to:
1) Override standard buttons, such as the New button for accounts, or the Save button for contacts.
2) Override tab overview pages, such as the Accounts tab home page.
3) Define custom tabs Embed components in detail page layouts.
4) Create dashboard components or custom help pages.[8]

VIII. APEX PROGRAMMING LANGUAGE

Force.com Apex Code is a strongly-typed programming language that executes on the Force.com platform. Apex is used to add business logic to applications, to write database triggers, and to program controllers in the user interface layer. It has a tight integration with the database and query language, good web services support, and includes features such as futures and governors for execution in a multi-tenant environment. Apex has a syntax that will be familiar to Java and C# programmers. It has the usual array of features such as classes, interfaces, constants, class variables, and annotations. Unusually, Apex is not case sensitive. Apex has strong ties with the Force.com persistence layer, the database. It shows how the language can be used to create, persist and update database objects (called sObjects), as well as query the database and iterate over results.[7] It is also used for writing database triggers. The Force.com platform supports two query languages:

- Salesforce Object Query Language (SOQL) is a query-only language. While similar to SQL in some ways, it's an object query language that uses relationships, not joins, for a more intuitive navigation of data.
- Salesforce Object Search Language (SOSL) is a simple language for searching across all persisted objects. These languages can be embedded in your Apex code, making data retrieval easy.

Apex as a controller language

Visualforce is the user interface layer on the Force.com platform. It provides a model-view-controller (MVC) paradigm to creating user interfaces, where incoming web requests (from a browser, say) can be routed to a controller that can perform certain actions, and then display a result.
Visualforce controllers and extensions must be written in Apex. These controllers typically look no different to the Apex code appearing in this article, though they do have access to additional data types, classes, methods and properties. For example, in Visualforce controllers you have access to page parameters, and Visualforce pages themselves are first-class citizens within Apex, allowin you to redirect to various pages, render the pages as PDF and email as a blob, and so on.[10]

XI. CONCLUSION

We have proposed a general approach for the generation of a visual editor for workflow rule. We have aimed at focusing on the characteristics of the visual syntax and visual editing, and thus separated the specification from the implementation of the user-interface of the editors. The work presented in this paper consists of our initial investigations at dealing with visual languages. In bridging the gap between workflow and design representations for interactive system development we have developed concepts and an environment for task oriented specification and prototyping. Designers are provided with a visual notation and graphical editors for the representation of workflows and their refinement to data, interaction and user models. The relationships between design elements and between the models themselves are automatically checked From our experiences thus far, we remain optimistic that our approach provides a good foundation for studying visual languages and programming and consequently the generation of visual editor for workflow rule.

REFERENCES

[1] B. Backlund. Visual programming languages and how to generate syntax-oriented environments for them. Technical Report TRITA-NA9OO3, Royal Institute of Technology, Sweden, 1990.

[2] B. Backlund, 0. Hagsand, and B. Pehrson. Generation of visual language-oriented design environments. Journal of Visual Languages and Computing,1~333-354, 1990.

[3] Jiří Tesařík, Ladislav Doležal and Christian Kollmann, "User interface design practices in simple single page web applications",2008 IEEE, Pg-223.

[4] Amin Roudaki, Mahmoud Doroodchi-"Interactive Software Development Using Workflow", 2009, International Conference on Future Computer and Communication,Pg.3-7.

[5] Schal, T. (1996),Workflow Management Systems for Process Organisations, Springer-Verlag, Berlin & Heidelberg, pp. 87, 89.

[6] Salesforce ,https://developer.force.com. org/

[7] Adding Programmatic logic with APEX, a workbook provided by Force.com.(August 2013 release).

[8] About the Visualforce workbook, a workbook provided by Force.com.(August 2013 release).

[9] [9]APEX workbook, a workbook provided by Force.com.(August 2013 release).

[10] wikipedia.org.com, (2008), available at http://en.wikipedia.org/wiki/Workflow , accessed on 20 Sept. 2013