APPLICATION OF USE CASE NOTATION FOR THE CONFIGURATION OF WORKFLOW ENGINES
TECHNICAL REPORT 607

AVNER OTTENSOOSER
SCHOOL OF IT, THE UNIVERSITY OF SYDNEY

DECEMBER, 2006
Application Of Use Case Notation For The Configuration Of Workflow Engines

Avner Ottensooser

Use case notation is generally used for requirements capture, while variants of the flowchart model (activity diagrams, Petri nets or similar models) are used to configure workflow engines. This position paper proposes that the Use Case notation can be an effective approach to configure workflow engines.

Reference:
Technical Report 607, School of Information Technologies, University of Sydney
Table of Contents

1. INTRODUCTION.................................................................................................................. 3
1.1. CONTEXT........................................................................................................................ 3
1.2. HYPOTHESIS.................................................................................................................. 4

2. WORKFLOW – THE CONTEXT OF THE POSITION PAPER........................................... 5

2.1. DEFINITION .................................................................................................................... 5
2.2. SEMANTIC ELEMENTS REQUIRED TO CONFIGURE A WORKFLOW ENGINE........ 6
2.3. CLEAR WF ENGINE CONFIGURATION – DEFINITION.................................................. 7
2.4. SEMANTICALLY CONSISTENT, WF ENGINE CONFIGURATION – DEFINITION........... 7
2.5. WORKFLOW NOTATIONS ............................................................................................... 8
   2.5.1. The Flowchart Model – Definition ........................................................................ 8
   2.5.2. The Flowchart Model in the WF context – Examples............................................ 9
   2.5.3. Clarity and semantic consistency of the Flowchart Model – Evaluation............... 9
2.6. PETRI NET NOTATIONS ............................................................................................... 10
2.7. BLOCK DIAGRAM NOTATIONS ................................................................................... 10
2.8. THE USE CASE NOTATION IN THE WORKFLOW CONTEXT ........................................ 10
   2.8.1. Topology ............................................................................................................. 10
   2.8.2. The Use Case Notation – examples .................................................................... 12
2.9. USE CASES, TRIGGERS, SCENARIOS, STEPS AND ACTORS ..................................... 14

3. CONFIGURING WORKFLOW WITH USE CASES.......................................................... 16

3.1. APPROACH ................................................................................................................... 16
3.2. USE CASE'S - TABULAR NOTATION .......................................................................... 16
3.3. APPLYING THE USE CASE NOTATION TO THE ALLOWED FLOW LAYER .................... 17
3.4. APPLYING THE USE CASE NOTATION TO THE USER PROFILE LAYER .................... 18
3.5. APPLYING THE USE CASE NOTATION TO THE USER INTERFACE LAYER ............... 19
3.6. APPLYING THE USE CASE NOTATION TO THE WORKFLOW STATE LAYER ............. 20
3.7. APPLYING THE USE CASE NOTATION TO THE AFTER THE EVENT DOCUMENTATION LAYER .................................................. 22
3.8. NORMALISATION CONCERNS .................................................................................... 23
3.9. SYNTAX ....................................................................................................................... 24

4. WORK NEEDED TO PROVE THE HYPOTHESIS............................................................. 25

SKELETON OF THE REFERENCE SECTION........................................................................ 26
1. Introduction

1.1. Context

Currently, some vendors of industrial workflow engines, such as IBM and TBICO deploy dialects of the Flowchart Model to configure their workflow engines (FileNET P8 and Staffware respectively). Other vendors use Petri Nets or Control Diagrams. These models lack clarity and semantic consistency. These deficiencies reduce the ability of the Workflow configuration officers to handle complexity.

To increase clarity and to simplify semantics, since 2002, BT Financial Group, a fully owned subsidiary of Westpac Banking Corporation, deployed the Use Case Notation using Cockburn’s dialect to configure its workflow engine.

“Clarity” and “Semantic consistency” are the pivotal anchors of this position paper. They will be used, in the WF context, to contrast the Use Case Notation with the other notations.

* Clarity – Can be interpreted in only one way

* Semantic Consistency – In this drafted PhD thesis context – The ability to use a single language to specify each of the four workflow elements Allowed Flow, Interface, State and History.
1.2. Hypothesis

This position paper generalises the lesson BT learned from deploying the Use Case Notation to configure its’ workflow engine. In particular, the Hypothesis that is presented tested and its implications evaluated is that:

1. Cockburn’s Use Case Dialect can be used as an input language to configure a workflow engine.

2. This engine will be able to handle certain workflow patterns.

3. Cockburn’s Use Case Dialect can be used as an input language to configure workflow engines in a way that is clearer then Flowcharts.

4. Cockburn’s Use Case Dialect can be used as an input language to configure workflow engines in a way that is more semantically consistent then Flowcharts.

By testing the Hypothesis, the position paper lays the theoretical foundations for the workflow industry to improve the clarity and the semantics consistency of its’ products, thus simplifying the role of workflow configuration officers and increasing the scalability of workflow engines (as complexity will be handled more efficiently).

As a structured language is described, the Use Cases can be fed into workflow engines using XML style syntax.
2. Workflow – the context of the position paper

2.1. Definition

As the definition of workflow varies, we quote a few supplementing definitions that will help a person new to the field to grasp the issue at hand. The definition is important because it is the starting point on which we build our arguments.

Workflow is…

- … a collection of steps and data that define the paths that can be taken to complete a task. Workflows may contain steps such as displaying content to users, collecting information from users or computer systems, performing calculations, and sending messages to external computer systems\(^4\).

- … a process description of how tasks are done, by whom, in what order and how quickly. Workflow can be used in the context of electronic systems or people, i.e., an electronic workflow engine can help automate a physician's personal workflow\(^5\).

- … an operational aspect of a work procedure: how tasks are structured, who performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are being tracked. As the dimension of time is considered in Workflow, Workflow considers "throughput" as a distinct measure. Workflow problems can be modelled and analysed using Petri nets\(^6\).
2.2. Semantic elements required to configure a workflow engine

Workflow Configuration articulates the following four aspects of the flow of work within an organisation:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Subject</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Allowed Flow</td>
<td>1. What are the allowed Steps in the system?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. How one identifies (indexes) a workflow item.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Under what circumstances should a work item flow from one Step to the other?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. At what order should work be delivered?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. What work can arrive to what server - what are the users’ privileges?</td>
</tr>
<tr>
<td>2</td>
<td>User Profile</td>
<td>How will the privileges and duties of users are defined</td>
</tr>
<tr>
<td>3</td>
<td>User Interface</td>
<td>How will the workflow engine interface with those who execute the Steps, whether they are human or machines.</td>
</tr>
<tr>
<td>4</td>
<td>State</td>
<td>What is the current state of a workflow item?</td>
</tr>
<tr>
<td>5</td>
<td>History</td>
<td>What path did a workflow item follow?</td>
</tr>
</tbody>
</table>
2.3. Clear WF engine configuration – Definition

A “clear” WF description model should be modular, unambiguous, reviewable and free of islands of duplication.

2.4. Semantically consistent, WF engine configuration – Definition

A semantically “consistent” WF description model should use a single syntax for all four aspects. Conversely, a semantically inconsistent WF description model would require a multiplicity of syntactical structures to configure them all.

The first of the two pivotal assertions of this position paper is that a workflow engine can be configured to infer the definition of all four layers from use cases.
2.5. Workflow Notations

2.5.1. The Flowchart Model – Definition

In its topologic core, the Flowchart Model has four constructs: Step, Decision-Construct, Border and Flow\(^*\). Accordingly, ISO standard 9004.4\(^8\) and ISO standard 5807\(^9\) use these four symbols\(^{10}\):

<table>
<thead>
<tr>
<th>Ovals are start/stop points.</th>
<th>Rectangles are Steps, processing, etc.</th>
<th>Diamonds are decision points.</th>
<th>Lines and arrows show the flow between the other symbols.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard to handle more than three conditions.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As the following exhibit demonstrates, by 1984 IBM enriched the Flowchart notation – but did not change its topology\(^{11}\).

According to the Flowchart model, work flows from one Step to the other, until completion. Each path and Step is graphically illustrated\(^{13}\).

\(^*\) A statement that we may need proving
2.5.2. The Flowchart Model in the WF context – Examples

The following three exhibits articulate how the WF engine configuration evolves as complexity is added, starting with flow 2.3.1 which is a sequence of two Steps, followed by sequence 2.3.2 where the second Step has two variants (“and this” and “or this”). In sequence 2.3.3 we introduce a loop.

2.5.3. Clarity and semantic consistency of the Flowchart Model – Evaluation

The following is a statement will be substantiated in the Position paper: As even the most basic real life business process is complex, has scores of exceptions the Flowcharts require A1 size printers\textsuperscript{14}. That sheer size deems the Flowcharts ambiguous\textsuperscript{15}, routing cannot be determined from its context. The Flowchart is neither maintainable\textsuperscript{16} nor reviewable\textsuperscript{17}. Nesting does not help, as it is rare to isolate an island of rules\textsuperscript{18}.
2.6. Petri Net Notations

A Petri Net is topologically similar to Flow Chart with the significant addition of state notation - the token. That is, Petri Net is a state full flow chart. As such it suffers from the same problems of flow charts.

2.7. Block diagram Notations

A Block Diagram is topologically more primitive then Flow Chart.. As such it suffers from the same problems of flow charts.

2.8. The Use Case Notation in the Workflow context

2.8.1. Topology

In its topologic core, the Use Case Notation has three constructs: Main Success Scenario Construct, Extensions Construct and Border Construct (Precondition, trigger). These constructs are supplemented by a protocol for the documentation of users and their Responsibilities (Actors).

Fundamental to the application of the Use Case Notation in the WF context is the concept of “Scenario”. A Scenario is initiated by a trigger and is composed of an ordered sequence of unconditional steps. A Scenario has no branching or alternatives. For each step a Role of a person or a machine that can perform the step is defined.

A Use Case is the collection of all the scenarios required to achieve a particular goal.

---

* The statement regarding the topologic nature of the use case model may be original.
* in a scenario there are no “IF” statements
* The granularity of the goal is a design consideration that should not be taken lightly. In the use case literature three means are prosed to control the Scenario explosion problem: subordinate use cases, extensions, and variations.
The Use Case Notation’s “Normal success scenario” construct represents the ordered list of steps required to process a work item. The “Extensions” section represents branches from the sequential sequence.

The following sections illustrate how the allowed flow of work from Step to Step is configured. To deploy the Use Case Notation on a FileNET workflow engine which was designed with the ‘Flowchart’ model in mind, BT Financial Systems configured its FileNET WF systems so that a workflow item may flow from any Role to any Role, in effect, creating a Use Case layer upon a Flowchart layer.

By doing so, complexity was not eliminated – it was migrated from a graph to narration. Fundamental to our position paper is our claim that the narrative notation of the Use Case is easier to comprehend than graphs. We intend to demonstrate that point using the software engineering literature.

Furthermore, as we deploy the Use Case Notation, we are better positioned to handle conditions with multiple alternative paths as the extension construct does not have the three ends of the flow chart diamond.
2.8.2. The Use Case Notation – examples

2.8.2.1. Use Case example 1 – Basic WF

Use Case 1 - Do something

Primary Responsibilities – A clerk

Main success scenario:

1. The clerk does some thing
2. The clerk does more

2.8.2.2. Use Case example – 2 … A Branch is added

Use Case 2 - Do something version 2

Primary Responsibilities – A clerk

Main success scenario:

1. The clerk does some thing
2. The clerk does more

Extensions:
1a The clerk failed to “do some thing”

1a.1 The clerk does some thing else
2.8.2.3. Use Case example – 3 … A Loop is added

Use Case 2 - Do something version 2

Primary Responsibilities – A clerk

Main success scenario

1. The clerk does some thing

2. The clerk does more

Extensions:
1a The clerk does failed to “do some thing”
   1a.1 The clerk does some thing else

2a more work is needed
   2a1 The clerk does repeats step 1
2.9. Use Cases, Triggers, Scenarios, Steps and Actors

In this section we articulate the constructs BT Financial Group deployed in order to apply the Use Case Notation to workflow configuration. BT Financial Group adopted an entity called Scenario. A Scenario is initiated by a trigger and is composed of Steps. For each step a Role of a person or a machine that can perform the step is defined. A Scenario indicates that, following a stated Trigger, a work item has to flow to certain Role bearers that may perform specified Ordered Steps on it.

A Scenario configuration includes:

- **Trigger** that models a condition that initiate the queuing of the work item. In general a trigger is an attribute of a WF item at a point in time, what someone sees now.

- **Ordered Steps** (Procedure) that model what someone has to do on the workflow item.

- **Responsibilities** that is the qualification of the user who will perform the Ordered Steps. In our domain, the term Responsibilities is more suitable then actor, as multi-skilled users have many Responsibilities.
For example, a Scenario can state that:

If a signature is missing (trigger),

Then Client Contact (Responsibilities)

Has to call the customer, and requesting the customer to sign the form (Ordered Steps)

A use case is the collection of scenarios available to resolve a distinct business problem (in BT’s case, the level of granularity adopted was that all the scenarios applicable, for example to Retirement Product Application, were hosted under the umbrella of a single use case).
3. Configuring Workflow with Use Cases

3.1. Approach

In this section we guide how to formalise use cases, written in natural language using the Cockburn style guide, with tabular style. Then we explain how the five WF engine documentation layers can be derived from these tables.

3.2. Use Case’s - Tabular Notation

A Use Case describes a business activity from the point of view of the primary actor who wishes to achieve some thing of benefit (a goal).

A use case is made of a main success scenario and optional extinction scenarios.

To initiate a scenario, one or many triggers are required.

A scenario may have one step or many ordered steps.
3.3. Applying the Use Case Notation to the Allowed Flow layer

The detachment of the Ordered Steps from triggers is required to execute a scenario from the initiating Triggers deskills the users. While Triggers change rarely, Steps are altered as the process evolves. To catalogue the Triggers, a category based tree structure was implemented.

At design time workflow administrators assign the scenarios (and indirectly and ordered list of Steps) to a triggers.

Scenario discovery is done with the help of traditional Use Cases. The “Use case” in its narrative form can be mapped one-to-one onto Triggers, Scenarios and Steps. The main success scenario is an ordered list\(^{19}\) of Steps. Each extension scenario is a triggered.

Triggers may, or may not, co-exist with other triggers. Examples,

<table>
<thead>
<tr>
<th>Trigger ↔ Trigger relationship</th>
<th>Business meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triggers co-existence:</td>
<td>A customer may sign a document, but may not date it.</td>
</tr>
<tr>
<td>Trigger exclusivity:</td>
<td>An application is either smaller then 100k, or grater/equal then 100k.</td>
</tr>
</tbody>
</table>
3.4. Applying the Use Case Notation to the User profile Layer

Users may be humans or subsystems. To record the ability of users to perform steps, the “Responsibilities” construct was created.

To increase the granularity of the despatching of Steps to Users, Difficulty was assigned to Steps and Skill was assigned to each responsibility. The WF engine may be configured to assign a Step to a user who is Responsible enough to handle the most difficult step triggered to that for the WF item*. 

* The dispatching of scenarios to users is outside the scope of this work.
3.5. Applying the Use Case Notation to the User interface layer

In this section we articulate what are the basic constructs BT Financial Group deployed to apply the Use Case Notation to the configuration of the workflow engine user interface. The users may be human or machines. At run time, users trigger scenarios. This activity is called piloting.

At every stage in the life cycle of the WF item, a user in his as a pilot is triggering certain scenarios. The user may also trigger (add diagnostic Triggers that may be translated by the engine into) instructions down stream.

At every state the user is presented with a list of Triggers to pick from. These become the entry criteria to another state (check box) and exit criteria from the current state.
3.6. Applying the Use Case Notation to the \textit{Workflow State} layer

In this section we articulate the basic construct we deployed in order enable a user to document in what state a workflow item is in.

The ability to retrieve routing sheets and WF items is a fundamental requirement from a workflow system. While external to the scope of this Position paper, we propose the use of an n-elements compound key. In BT’s case three elements where elected.

\begin{center}
\textit{Indexing structure}
\end{center}

At run time a new data entity is needed – Work Flow Item (workflow Item).

A workflow has several Scenarios modelling the specific flows the workflow item may follow within the organisation.

At run time a new data entity is needed – supporting document.
Traditionally a supporting model was an image, which was interchangeable with workflow item. Today the supporting document may be an XML document as well.

The act of triggering steps to a workflow item is dubbed “Piloting”. It is initially done as part of the indexing of a workflow item. Privileged users may trigger steps to any open Workflow item at any time.

At run time the workflow engines may assign the oldest unpiloted work, then the oldest work and the hardest work item a user may perform from the oldest day.
3.7. Applying the Use Case Notation to the After the Event documentation layer

In this section we articulate what are the basic construct we deployed in order to record the historic flow of work through the organisation.

The Use Case WF Configuration language model enables event-based communication with an external reporting system using the following three message types:

<table>
<thead>
<tr>
<th>Message</th>
<th>Sent when</th>
<th>Describe</th>
<th>Used to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow item</td>
<td>a workflow is created or terminated</td>
<td>Indexing information.</td>
<td>Calculate external SLAs</td>
</tr>
<tr>
<td>Triggers Message</td>
<td>a Scenario is assigned to a workflow item</td>
<td>Attributes of a workflow item.</td>
<td>Triggers Messages are important for QA.</td>
</tr>
<tr>
<td>Step Message</td>
<td>is queued, start, end, or diarised.</td>
<td>Work performed on workflow item (Who performed the Step?)</td>
<td>Internal SLAs</td>
</tr>
</tbody>
</table>
3.8. Normalisation concerns

Scenarios are artificial constructs that resolve the many to many relationships between Work type and Step. The Scenarios lists the Steps that may be performed on a work type as well as the proficiency required from the user who performs that specific Step (skill level).

To group Triggers into human comprehensible clusters, we have introduced a classification mechanism named category – which is presented as a directory structure.

Allowed Steps are artificial constructs that resolve the many to many relationships between User and Step. The Allowed Steps of a user is a list of all the Steps a user may perform, as well as the users’ proficiency in performing each of these Steps (skill level).
3.9. Syntax

To write the use case the following three syntactical elements are required:

<Role>

<User> <Role> <skill>

<Sequence factor> <trigger> [extending <Sequence factor>]

<Sequence factor> [extend by] [<Role> <Step> <difficulty> [and <Role> ,<Step>, <difficulty>]]

<Sequence factor> Return [to Person] or [Same Role other person]

In BT’s case, rather then write an interpreter, BT instructed the workflow administrators to write into BT’s database using a dedicated table driven user interface.
4. Work needed to prove the Hypothesis

We plan to prove the Hypothesis using the following techniques:

<table>
<thead>
<tr>
<th>Claim</th>
<th>Proof Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Completeness of the model</td>
<td>Review of a rich library of patterns will show us if the use case notation is sufficiently generic. A library of patterns is maintained at <a href="http://is.tm.tue.nl/research/patterns/">http://is.tm.tue.nl/research/patterns/</a></td>
</tr>
<tr>
<td>2 Practicality of the model</td>
<td>Demonstrate how an industrial workflow system is currently being fed with use cases (albeit in its tabular form).</td>
</tr>
<tr>
<td>3 Clarity of the model</td>
<td>Usability experiment can demonstrate that one model is clearer than another.</td>
</tr>
<tr>
<td>4 Clarity of the model</td>
<td>We also plan to prove this assertion by referring to the research done over the advantages of use case for process documentation.</td>
</tr>
<tr>
<td>5 Semantic Consistency of the model</td>
<td>By elaborating section 3 - CONFIGURING WF WITH USE CASES</td>
</tr>
</tbody>
</table>
SKELETON OF THE REFERENCE SECTION

Structuring use cases with goals - Alistair A.R. Cockburn -
http://alistair.cockburn.us/index.php/Structuring_use_cases_with_goals

ISO/IEC JTC1/SC7/WG11 the Petri net ISO standard

An overview of workflow management: From process modeling to workflow automation infrastructure Journal Distributed and Parallel Databases Publisher Springer Netherlands ISSN 0926-8782 (Print) 1573-7578 (Online) Subject Computer Science Issue Volume 3, Number 2 / April, 1995 DOI 10.1007/BF01277643 Pages 119-153 Online Date Thursday, March 31, 2005


1 Research note – place a quote to the FileNET P8 manual.

2 Research note – place a quote to the TIBCO Staffware manual.

3 Alistair Cockburn, Writing Effective Use Cases, Addison-Wesley, 2001, pp 132

4 www.scbos.com/Info/SCBOS-Site-Glossary.htm

5 www.thebusinesssite.com/IT%20Terms/Health%20Terms.htm

6 en.wikipedia.org/wiki/Workflow

7 The information covered in this section is explained in detail by Tom Baeyens at http://www.theserverside.com/articles/article.tss?f=Workflow

8 Research note – put a clear reference to ISO 9004.4

9 Research note – put a clear reference to ISO 5807
10 http://www.georgehernandez.com/h/xzMisc/UML/Flowchart.htm

11 Research note – the statement that IBM enriched the model vocabulary but did not change its topological nature may need proving in the drafted thesis.


13 Research note – find a clear quote from ISO 9004.4 or from ISO 5807 to demonstrate this.

14 Research note – this will require empiric evidence from the industry.

15 Research note – the ambiguity would require either usability test or mathematical demonstration.

16 Research note – the maintainability would require usability test.

17 Research note – the reviewability would require usability test.

18 Research note – find quotes from the 70s regarding the advantage of procedural modelling.

19 Additional patterns are described Prof Wil van der Aalst at al in: http://is.tm.tue.nl/research/patterns/index.htm