WORKSPACES

Design and Implementation of a distributed Workflow System based on XML / XSL

Presentation of current Work in Progress

York, December 1999

Marc Stauch

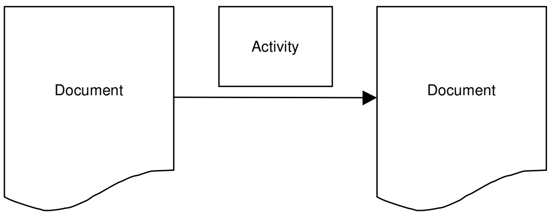
Technical University of Berlin

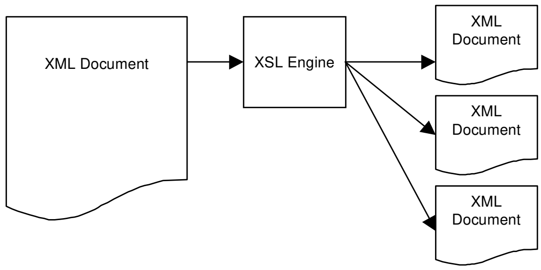
marc@stau.ch http://www.cs.tu-berlin.de/~stauch/DIPLOM/

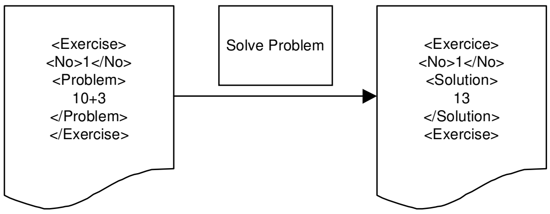
Workspaces

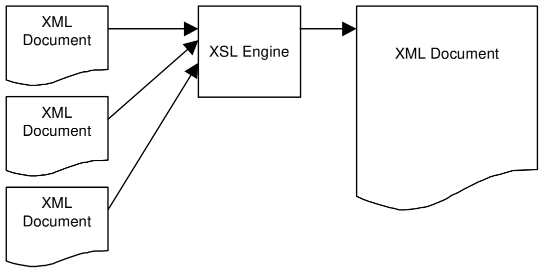
Design and Implementation:

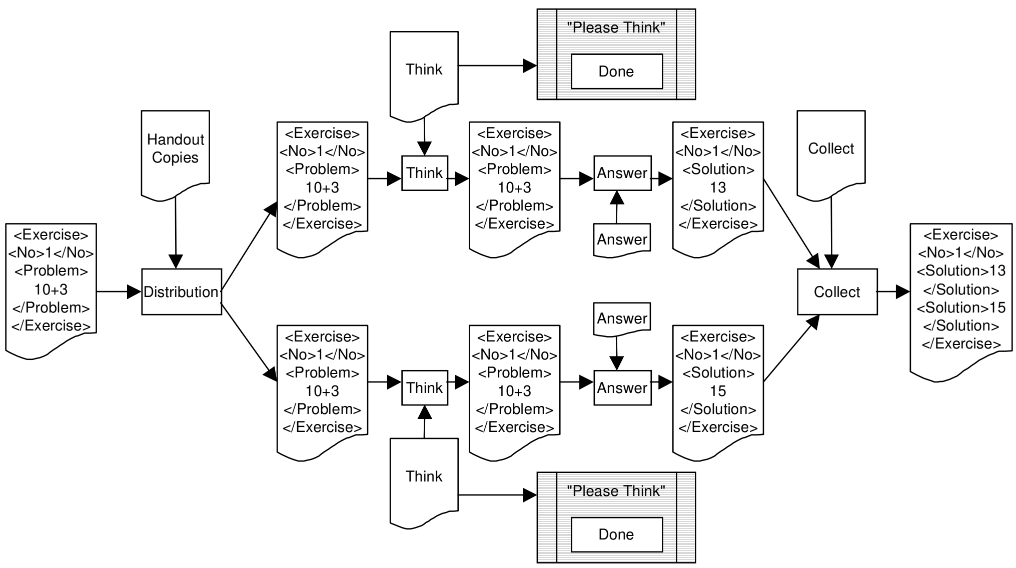
- A Document Driven Workflow
- The Concept of Steps
  - External, User, Automatic Steps
  - Coordination Steps
    - Join
    - Split
    - Loop


Document
Activity
Document


XML Document
XSL Engine
XML Document
XML Document
XML Document

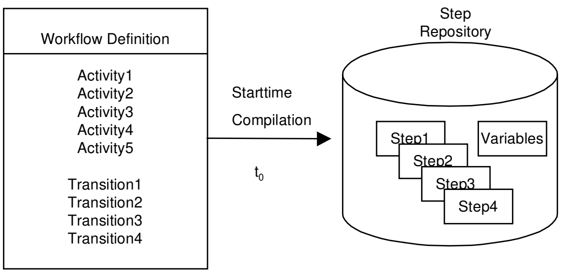

<Exercise> <No>1</No> <Problem> 10+3 </Problem> </Exercise>
Solve Problem
<Exercice> <No>1</No> <Solution> 13 </Solution> <Exercise>

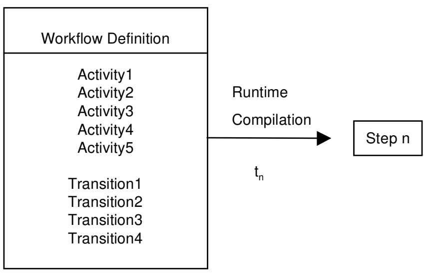

XML Document
XML Document
XML Document
XSL Engine
XML Document

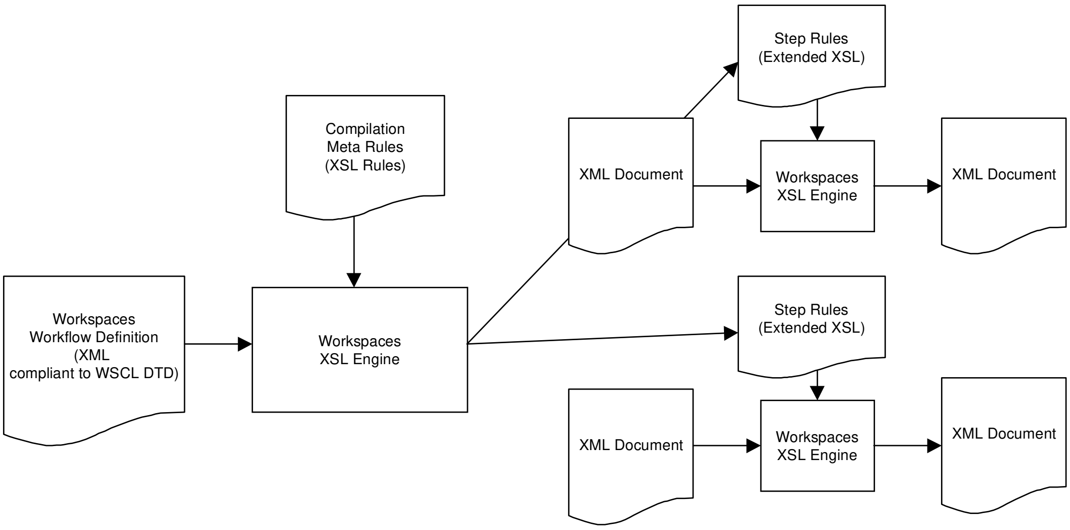

Think
"Please Think"
Done
Handout Copies
<Exercise> <No>1</No> <Problem> 10+3 </Problem> </Exercise>
Distribution
<Exercise> <No>1</No> <Problem> 10+3 </Problem> </Exercise>
Think
<Exercise> <No>1</No> <Problem> 10+3 </Problem> </Exercise>
Answer
Answer
<Exercise> <No>1</No> <Solution> 13 </Solution> </Exercise>
Collect
Collect
<Exercise> <No>1</No> <Solution>13 </Solution> <Solution>15 </Solution> </Exercise>
<Exercise> <No>1</No> <Problem> 10+3 </Problem> </Exercise>
Think
<Exercise> <No>1</No> <Problem> 10+3 </Problem> </Exercise>
Answer
Answer
<Exercise> <No>1</No> <Solution> 15 </Solution> </Exercise>
Think
"Please Think"
Done

XML / XSL:

- XML Documents
  - XML Space Coordination Medium
    - Linda-like, T-Spaces
    - User Dependant Basic Operations
      - Read
      - In
      - Out
- XSL Step Compilation Rules
  - Workflow, Step Definition
  - Meta Rules (Compilation of Steps)
  - XSL Engine

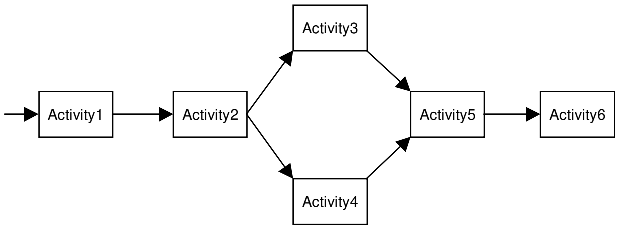


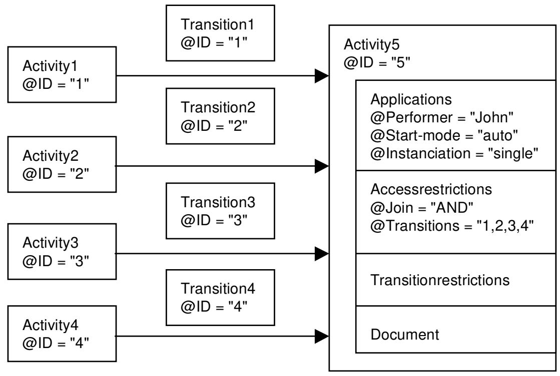


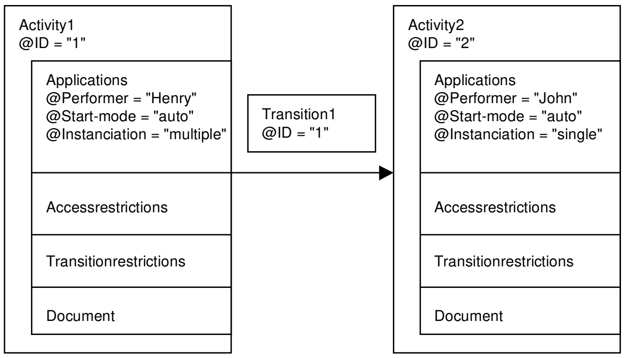


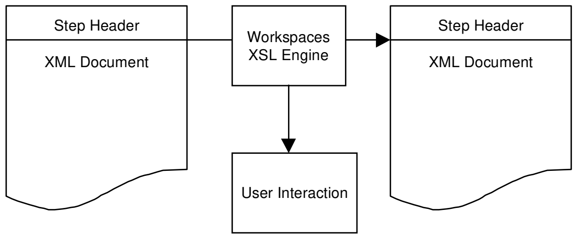
Design Issues:

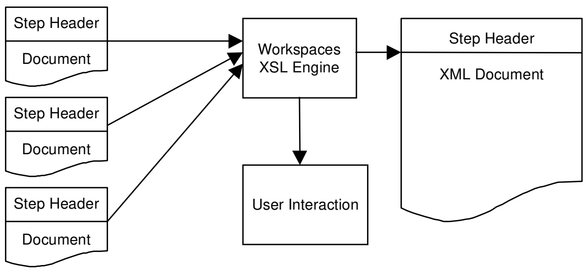
- <u>Workflow DTD</u>
  - Using URLs
- Step Compilation
  - Lazy (Runtime) vs. Eager (Starttime)
  - Necessary Step Information
- Step Execution
  - Basic Steps
  - Coordination Steps
- Step DTD
  - Eager vs. Lazy, Repository

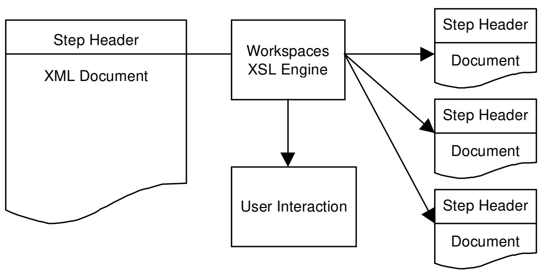

Activity1
Activity2
Activity3
Activity4
Activity5
Activity6

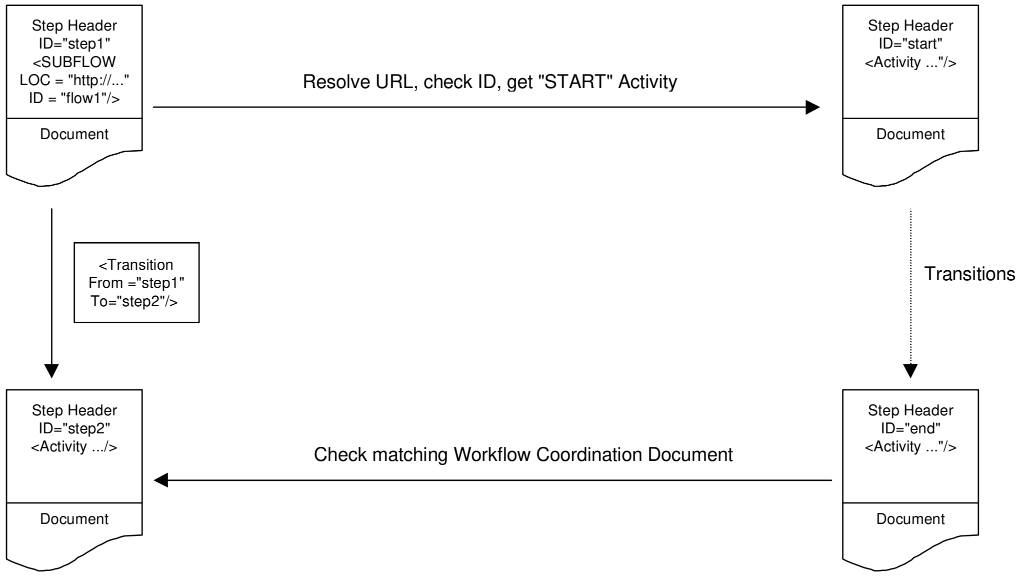

Transition1
@ID = "1"
Activity1
@ID = "1"
Transition2
@ID = "2"
Activity2
@ID = "2"
Transition3
@ID = "3"
Activity3
@ID = "3"
Transition4
@ID = "4"
Activity4
@ID = "4"
Activity5
@ID = "5"
Applications
@Performer = "John"
@Start-mode = "auto"
@Instanciation = "single"
Accessrestrictions
@Join = "AND"
@Transitions = "1,2,3,4"
Transitionrestrictions
Document

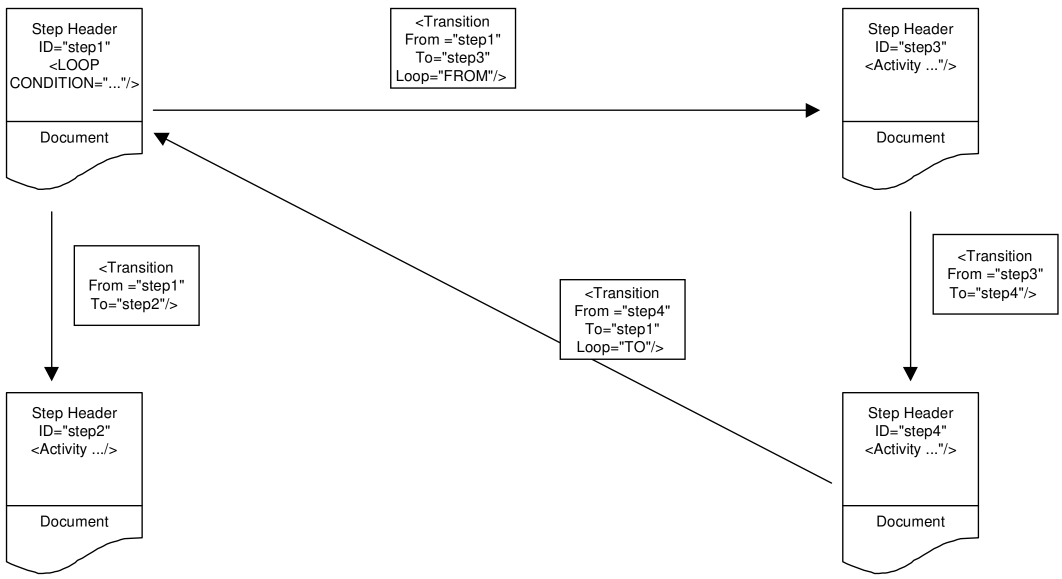

Activity1
@ID = "1"
Applications
@Performer = "Henry"
@Start-mode = "auto"
@Instanciation = "multiple"
Accessrestrictions
Transitionrestrictions
Document
Transition1
@ID = "1"
Activity2
@ID = "2"
Applications
@Performer = "John"
@Start-mode = "auto"
@Instanciation = "single"
Accessrestrictions
Transitionrestrictions
Document


Step Header
XML Document
Workspaces
XSL Engine
Step Header
XML Document
User Interaction


Step Header
Document
Step Header
Document
Step Header
Document
Workspaces
XSL Engine
Step Header
XML Document
User Interaction

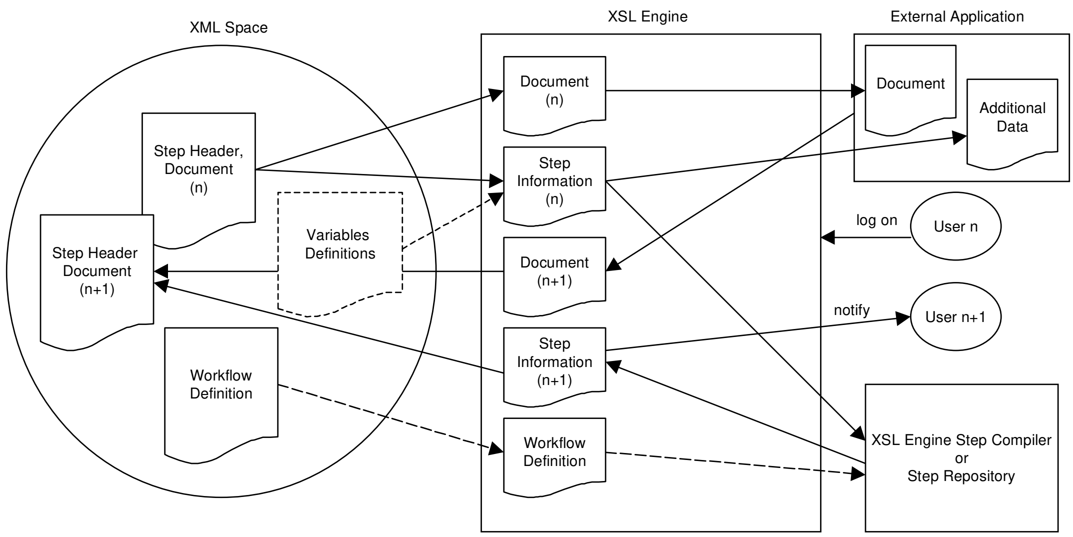

Step Header
XML Document
Workspaces
XSL Engine
User Interaction
Step Header
Document
Step Header
Document
Step Header
Document

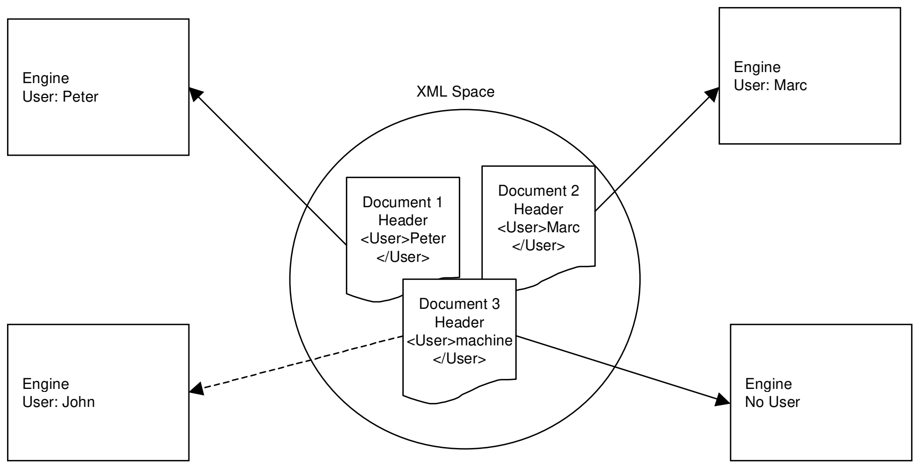

Step Header
ID="step1"
<SUBFLOW
LOC = "http://..."
ID = "flow1"/>
Document
Resolve URL, check ID, get "START" Activity
Step Header
ID="start"
<Activity ..."/>
Document
<Transition
From ="step1"
To="step2"/>
Transitions
Step Header
ID="step2"
<Activity .../>
Document
Check matching Workflow Coordination Document
Step Header
ID="end"
<Activity ..."/>
Document

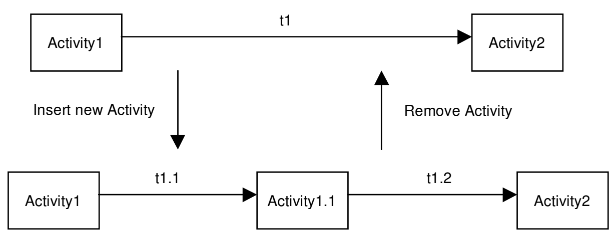

Step Header
ID="step1"
<LOOP
CONDITION="..."/>
Document
<Transition
From ="step1"
To="step3"
Loop="FROM"/>
Step Header
ID="step3"
<Activity ..."/>
Document
<Transition
From ="step1"
To="step2"/>
<Transition
From ="step4"
To="step1"
Loop="TO"/>
<Transition
From ="step3"
To="step4"/>
Step Header
ID="step2"
<Activity .../>
Document
Step Header
ID="step4"
<Activity ..."/>
Document

Design Issues:

- The Engine
  - Basic Characteristics
    - Eager vs. Lazy
  - Coordination of Multiple Engines
- Meta Steps - Updating Workflows
  - Eager vs. Lazy, Backups
- User Preferences
- Working Offline




XML Space
Engine
User: Peter
Engine
User: Marc
Engine
User: John
Engine
No User
Document 1
Header
<User>Peter
</User>
Document 2
Header
<User>Marc
</User>
Document 3
Header
<User>machine
</User>


Activity1
t1
Activity2
Insert new Activity
Remove Activity
Activity1
t1.1
Activity1.1
t1.2
Activity2

Design Issues:

- XML Space
  - Basic Operations:
    - Read (Key -> List of Strings)
    - In (Key, String -> Step as DOM)
    - Out (Step as DOM)
    - Copy (extracting Subtrees)
  - Consistency, Completeness
  - Distributed
  - Not Real-Time, Not Blocking
  - Security, Safety
    - Permissions, Groups
    - Step Restoration

Implementation:

- Tools
  - XML: DOM (IBM's XML4J)
  - XSL: James Clark's XT
  - Java: jdk1.1.2
  - Iava (Java Interpreter)
- GUI
- Step Compilation (Lazy)
- Resolving Variables
- Coordination Mechanisms

Outlook:

- Graphical Interface for the Development of Workflow Definitions
- Implementation of XML Space
  - Distribution
  - Mobility
  - Safety
- Running Workspaces Offline
  - Extracting Subtrees of XML Space
  - Resolving External URL References
- Safety, Security
- Eager Compilation of Steps

EXAMPLE OF A WORKFLOW DEFINITION

```
<?xml version="1.0"?>
<!DOCTYPE MODEL SYSTEM "wfcl.dtd">
<MODEL ID="model1" NAME="TestModel">
<!-- participants -->

<!-- data -->

<DATA ID="P1" WORKFLOW="wkfl2" TYPE="dummy"><PARTICIPANT ID="dummy" MAIL="marc@stau.ch" NAME="Marc Stauch" TYPE="HUMAN"/></DATA>
<DATA ID="P1" WORKFLOW="wkfl1" TYPE="dummy"><PARTICIPANT ID="dummy" MAIL="marc@stau.ch" NAME="Marc Stauch" TYPE="HUMAN"/></DATA>
<DATA ID="P2" WORKFLOW="wkfl1" TYPE="dummy"><PARTICIPANT ID="dummy" MAIL="stauch@cs.tu-berlin.de" NAME="Marc Stauch" TYPE="HUMAN"/></DATA>
<DATA ID="P3" WORKFLOW="wkfl1" TYPE="dummy"><PARTICIPANT ID="dummy" TYPE="SYSTEM"/></DATA>

<APPLICATION
 ID="editor"
 TOOLNAME="editor"/>

<!-- workflows -->
<WORKFLOW ID="wkfl1" LOC=“file//:\Engine\XMLSpace\Workflows\kurs.xml" NAME ="TEST Workflow" DOCUMENTATION="http://www.cs-tu-berlin.de/~stauch/DIPLOM/wkfl1.html">

<PARTICIPANT PARAM="P1" ID="myself" TYPE="HUMAN"/>
<PARTICIPANT PARAM="P2" ID="other" TYPE="HUMAN"/>
<PARTICIPANT PARAM="P3" ID="automatic" TYPE="SYSTEM"/>

<ACTIVITY
 ID="start"
 NAME ="First Task wkfl1">
<DESCRIPTION>Description of the first Step of the first Workflow. The following two Steps are Split with a SPLIT XOR restriction.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
         INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
<TRANSITIONRESTRICTION SPLIT="XOR" TRANSITIONS="ID_1,ID_2"/>
</ACTIVITY>

<ACTIVITY
 ID="ID_1.2"
 NAME ="Second Task wkfl1">
<DESCRIPTION>This is the second Step of the first Workflow. It is one of the two XOR Split alternatives.</DESCRIPTION>
<APPLICATIONS PERFORMER="other"
         DOCUMENTATION="http;//www.cs-tu-berlin.de/~stauch/DIPLOM/StepID_1.2.html"
         INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>
```

```xml
<ACTIVITY
 ID="ID_1.3"
 NAME ="Third Task wkfl1">
<DESCRIPTION>This is the third step of the first Workflow. It is the second of the two XOR Split alternatives (Defined by the transitionrestriction Element)</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
          INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>


<ACTIVITY
 ID="ID_1.4"
 NAME ="Fourth Task wkfl1">
<DESCRIPTION>This is the forth step of the first Workflow. It automatically joins the two previous alternatives with a JOIN XOR transitionrestriction. Hence you will never see this description
        text when running the workflow.</DESCRIPTION>
<APPLICATIONS PERFORMER="automatic"
          INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
<TRANSITIONRESTRICTION JOIN="OR" TRANSITIONS="ID_4,ID_3"/>
</ACTIVITY>

<ACTIVITY
 ID="ID_1.5"
 NAME ="Fifth Task wkfl1">
<DESCRIPTION>This is the fifth Step of the first Workflow. it is really an automatic Step that executes the second Workflow, this Workflow can be defined within this same Workflow
        definition file or externally in a Workflow Definition URI.</DESCRIPTION>
<SUBFLOW ID="wkfl2"/>
</ACTIVITY>

<ACTIVITY
 ID="end"
 NAME ="Final Task wkfl1">
<DESCRIPTION>This is the last Step of the first Workflow. The file this Step produces is terminal.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
          INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>
```

```xml
<TRANSITION ID="ID_1"
        FROM="start"
        TO="ID_1.3"/>
<TRANSITION ID="ID_2"
        FROM="start"
        TO="ID_1.2"/>
<TRANSITION ID="ID_3"
        FROM="ID_1.3"
        TO="ID_1.4"/>
<TRANSITION ID="ID_4"
        FROM="ID_1.2"
        TO="ID_1.4"/>
<TRANSITION ID="ID_5"
        FROM="ID_1.4"
        TO="ID_1.5"/>
<TRANSITION ID="ID_5"
        FROM="ID_1.5"
        TO="end"/>
</WORKFLOW>

<WORKFLOW ID="wkfl2" LOC="D:\Engine\XMLSpace\Workflows\kurs.xml" NAME ="TEST Workflow" DOCUMENTATION="http://www.cs.tu-berlin.de/~stauch/DIPLOM/wkfl2.html">

<PARTICIPANT PARAM="P1" ID="myself" TYPE="HUMAN"/>

<ACTIVITY
 ID="start"
 NAME ="First Task WKFL2">
<DESCRIPTION>This is the first step of the second (SUB)Workflow, it is followed by an AND SPLIT.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
          INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
<TRANSITIONRESTRICTION SPLIT="AND" TRANSITIONS="ID_1,ID_2"/>
</ACTIVITY>

<ACTIVITY
 ID="ID_1.1"
 NAME ="Second Task WKFL2">
<DESCRIPTION>This is the second step of the second (SUB)Workflow. Executed at the same time as the third step.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
          INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>
```

```xml
<ACTIVITY
 ID="ID_1.2"
 NAME ="Third Task WKFL2">
<DESCRIPTION>This is the third step of the second (SUB)Workflow. Executed at the same time as the second step.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
            INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>

<ACTIVITY
 ID="ID_1.3"
 NAME ="Join Task WKFL2">
<DESCRIPTION>This is the last step of the second (SUB)Workflow.
Subsequently return to the sixt step of the fist Workflow. It unites the two previous steps using JOIN AND Transitionestriction.
It will be followed by a LOOP Step</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
            INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
<TRANSITIONRESTRICTION JOIN="AND" TRANSITIONS="ID_3,ID_4"/>
</ACTIVITY>

<ACTIVITY
 ID="ID_1.4"
 NAME ="Start LOOP Task WKFL2">
<DESCRIPTION>This is a LOOP Step within the second (SUB)Workflow. The LOOP is executed until an Element DONE will be found within the document that contains the word
          'true'</DESCRIPTION>
<LOOP KIND="while" CONDITION='(!workspaces.Conditions.element("DONE").equalsIgnoreCase("true"))'/>
</ACTIVITY>

<ACTIVITY
 ID="ID_1.5"
 NAME ="Loop BODY WKFL2">
<DESCRIPTION>This is the body step of the LOOP within the second (SUB)Workflow.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
            INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>
```

```xml
<ACTIVITY
 ID="end"
 NAME ="Last Task WKFL2">
<DESCRIPTION>This is the last step of the second (SUB)Workflow.
Subsequently we will return to the sixt step of the fist Workflow. Once we are here, the LOOP is finished.</DESCRIPTION>
<APPLICATIONS PERFORMER="myself"
            INSTANTIATION="multiple">
<TOOL NAME="editor"/>
</APPLICATIONS>
</ACTIVITY>

<TRANSITION ID="ID_1"
          FROM="start"
          TO="ID_1.1"/>

<TRANSITION ID="ID_2"
          FROM="start"
          TO="ID_1.2"/>

<TRANSITION ID="ID_3"
          FROM="ID_1.2"
          TO="ID_1.3"/>

<TRANSITION ID="ID_4"
          FROM="ID_1.1"
          TO="ID_1.3"/>

<TRANSITION ID="ID_8"
          FROM="ID_1.3"
          TO="ID_1.4"/>

<TRANSITION ID="ID_5"
          FROM="ID_1.4"
          TO="ID_1.5"
          LOOP="FROM"/>

<TRANSITION ID="ID_6"
          FROM="ID_1.5"
          TO="ID_1.4"
          LOOP="TO"/>

<TRANSITION ID="ID_7"
          FROM="ID_1.4"
          TO="end"/>
</WORKFLOW>
</MODEL>
```

Example of a generated Step

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP>
<WORKFLOW DOCUMENTATION="http;//www.cs-tu-berlin.de/~stauch/DIPLOM/wkfl1.html"
    ID="wkfl1" LOC="D:\Engine\XMLSpace\Workflows\kurs.xml" NAME="TEST Workflow"/>
<NAME>Second Task wkfl1</NAME>
<ID>ID_1.2</ID>
<DOCUMENTATION></DOCUMENTATION>
<DATE>19991201053419000</DATE>
<DESCRIPTION>This is the second Step of the first Workflow. It is one of the two XOR Split
    alternatives.</DESCRIPTION>
<RESTRICTION mode="XOR" type="SPLIT">
<FROM>start</FROM>
<TO>ID_1.3</TO>
<FROM>start</FROM>
<TO>ID_1.2</TO>
</RESTRICTION>
<TYPE>External</TYPE>
<INSTANTIATION>multiple</INSTANTIATION>
<PARTICIPANT ID="other" MAIL="stauch@cs.tu-berlin.de" NAME="Marc Stauch"
    TYPE="HUMAN"/>
<APPLICATION ID="editor" TOOLNAME="editor"/>
<DOCUMENT><TEST>
Content of the attached Document</TEST>
</DOCUMENT>
</STEP>
```

EXAMPLE OF AN XSL Style Sheet

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
            xmlns:xt="http://www.jclark.com/xt"
                    version="1.0"
            extension-element-prefixes="xt">

<!-- RESOLVE Variables in Workflows -->
<xsl:template match="*[@PARAM]" priority="1">
        <xsl:variable name="wkfl" select="ancestor::WORKFLOW/@ID"/>
        <xsl:variable name="id" select="@ID"/>
        <xsl:variable name="param" select="@PARAM"/>
        <xsl:choose>
                    <xsl:when test="/child::MODEL/child::DATA[@ID=$param][@WORKFLOW=$wkfl]/child::*">
<xsl:variable name="tag" select="name(/child::MODEL/child::DATA[@ID=$param][@WORKFLOW=$wkfl][position()=1]/child::*[position()=1])"/>
                                        <xsl:element name="{$tag}">
                                        <xsl:for-each
        select="/child::MODEL/child::DATA[@ID=$param][@WORKFLOW=$wkfl][position()=1]/child::*[position()=1]/@*">
                                                    <xsl:variable name="att" select="name(.)"/>
                                                    <xsl:attribute name="{$att}">
                                                    <xsl:value-of select="."/>
                                                    </xsl:attribute>
                                        </xsl:for-each>
                                        <xsl:attribute name="ID">
                                                    <xsl:value-of select="$id"/>
                                        </xsl:attribute>
                                        </xsl:element>
                    </xsl:when>
                    <xsl:otherwise>
                                        <xsl:comment>could not resolve variable</xsl:comment>
                                        <xsl:copy>
                                          <xsl:apply-templates select="@*|*|text()"/>
                                        </xsl:copy>
                    </xsl:otherwise>
        </xsl:choose>
</xsl:template>
```

```
<xsl:template match="*|@*" priority="-1">
  <xsl:copy>
    <xsl:apply-templates select="@*|*|text()"/>
  </xsl:copy>
</xsl:template>
```