

Workflow Requirements Modelling Using XML

Bill Karakostas^a, Dimitris Panagiotakis^a and George Fakas^b

^aCentre for HCI Design, School of Informatics, City University, London, UK; ^bDepartment of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK

When modelling inter-organisational workflow it is important not to make assumptions such as with regard to the formats of the data exchanged between the workflow participants or the technical infrastructures and platforms, as they can restrict the range of possible workflow management implementations. The approach presented in this paper allows for the conceptual modelling of workflow processes using primitive constructs such as nodes, rules and business documents. The paper presents both a graphical notation for modelling workflows as well as a mapping of the workflow constructs to XML models that follows the Workflow Management Coalition interoperability standards. This allows the modelled workflow to be interpreted and executed by a variety of workflow engines.

Keywords: WfMC; Workflow management systems; XML; XPDL

1. Introduction

Productivity and customer service are key business issues for the e-businesses of today. They are both inextricably linked to the technology of workflow management software. Forecasts show a rapid increase in sales of workflow management software over the next few years [1].

Workflow is the name given to procedures that involve the routing of tasks from person to person in sequence, allowing each to make a contribution before moving it on to the next stage. Examples are the processing of a business order requisition, or an

insurance claim. Increased productivity in workflow systems comes from automated support of storing, creating, accessing, processing and routing of forms, letters and so on.

With the rapid growth of e-business, inter-organisational workflow, i.e. the ability to manage workflow processes that span several organisations, is becoming more important. In response to that, the Workflow Management Coalition (WfMC) is currently defining a simple protocol that enables interaction between requesters and providers of workflow-type services [2].

In inter-organisational workflow interoperability of workflow data becomes more of an issue. In general, different organisations use a variety of proprietary data formats and workflow execution engines. Thus, workflow exchanges must be modelled on a technology-neutral and universal protocol ideally based on the Extensible Markup Language (XML).

Our approach views workflow as the *flow* of business relevant *knowledge* encoded in business documents, within and across an organisation's boundaries. A workflow process is defined as a set of *workflow execution nodes* and *rules*, which interconnect to define the flow of the business document within or outside the organisation. We have defined a simple graphical modelling formalism based on a small set of primitives that allows workflow requirements to be described without making assumptions as to specific data formats and workflow execution engines. To ensure interoperability of workflows across different systems we encode business documents in XML. In addition, the whole workflow process can be defined as an XML schema that can be interpreted at run time by a workflow engine.

The structure of this paper is as follows. Section 2 reviews related approaches to workflow modelling that use UML, XML and other notations. Section 3 describes the different types of workflow catered by our approach,

Correspondence and offprint requests to: B. Karakostas, Centre for HCI Design, School of Informatics, City University, Northampton Square, London EC1V 0HB, UK. Email: billk@soi.city.ac.uk

while Section 4 introduces our workflow meta-model. Section 5 illustrates the proposed workflow modelling approach using the case study of an electronic publisher. Section 6 summarises our approach and discusses further research.

2. Workflow Modelling Approaches

2.1. The Workflow Management Coalition (WfMC) Meta-Model

The WfMC Process Definition meta-model identifies the result of the build-time operation, i.e. the process definition, as one of the potential areas of standardisation to enable the interchange of process definition data between different build-time tools and run-time products. The Coalition has published a draft for a 'Workflow Process Definition Language' that attempts to unify the different modelling approaches of the workflow management systems and Business Process Re-engineering tools [3].

2.2. Other Workflow Modelling Approaches

A variety of modelling notations have been used to specify workflow at a conceptual or design level. Adam et al. [4] present a Petri net-based framework that according to the authors facilitates specification of workflow applications, serves as a powerful tool for modelling the system under study at a conceptual level, allows for a smooth transition from the conceptual level to a tested implementation and enables analysis, simulation and validation of the system under study before processing to implementation.

Wagner et al. [5] present a workflow approach of information systems development (WFSAD: Workflow System Analysis and Design) that incorporates an emphasis on analysing business processes to improve processing time and focuses on developing integrated systems throughout and beyond organisational boundaries. A framework for analysing the impact of the WFSAD approach on the management of information systems personnel is also presented.

UML is a standard language for specifying, visualising, constructing and documenting the artefacts of software systems as well as for business modelling and other non-software systems. UML has recently become a popular notation for the specification of workflow. Hruby [6] demonstrates how UML can be used for specification of workflow management systems, how to trace the description of business processes to the object-oriented software design and how to structure the project repository with UML deliverables. In this modelling

approach, business objects are represented by classes and instances. Business processes are represented by use cases and use case instances. Workflows are automated business processes. They are represented as use cases or use case instances with a stereotype 'workflow'. Team roles are represented by classes and objects in UML. Classes represent types of team roles; objects represent concrete workers playing the role.

The UML Revision Task Force is in the process of defining extensions to activity diagrams to address expressiveness problems [7]. According to Schmidt [8], when this work is completed UML workflow models could become a base for generation of workflow components and the integration of other business components into workflows.

2.3. The Role of XML in Workflow Management

The Workflow management Coalition (WfMC) has proposed the use of XML as the mechanism for process definition interchange. XPDL (XML Process Definition Language) forms a common interchange standard that enables products to continue to support arbitrary internal representations of process definitions with an import/export function to map to/from the standard at the product boundary.

One of the key elements of XPDL is its extensibility to handle information used by a variety of different tools. Based on a limited number of entities that describe a workflow process definition (the 'Minimum Meta Model'), XPDL supports a number of different approaches. A variety of different mechanisms may be used to transfer process definition data between systems according to the characteristics of the various business scenarios. In all cases the process definition must be expressed in a consistent form, which is derived from the common set of objects, relationships and attributes expressing its underlying concepts.

3. Types of Workflow

At the highest-level workflow can be divided into the following categories according to the way the process spreads through the organisation or outside its boundaries.

Local workflow. Local workflow is performed by a specific user and does not involve a business document being sent to another user. It is essentially a sequence of tasks that a user can perform under a certain role in order to achieve a goal.

Internal (organisational) workflow. Internal workflow represents workflow processes that are taking place within the organisation.

External workflow. External workflow represents processes that span out of the organisation. For this kind of workflow, there are process integration issues between the different organisations.

At a lower level, workflow processes can be broken down as follows:

Predefined workflow. Predefined workflow is specified as part of the organisational modelling and represents core processes of the organisation.

Predefined workflow can be further broken down into:
Administrative workflow. Administrative workflow represents the core business processes that automate routine approval processes such as document authorisation and expense report processing. In administrative workflow the content of the business document usually stays unchanged. In administrative workflow the instance of a business document resides in one node only at any one time.

Production workflow. Production workflow represents the core business processes that provide a profit. Examples include purchasing and sales. In production workflow the business document may change or even transform to another business document along the process.

Collaborative workflow. Collaborative workflow represents the core business processes that require groups of people to work as a team in order to achieve a goal (produce or edit a business document). In these processes the business document contents will most likely change. In collaborative workflow an instance of a business document may reside in more than one node at the same time, being edited in parallel.

Predefined workflow can also be broken down to automated or manual according to the way it is initiated.

- *Automated.* Automated workflow processes start by system events rather than explicit user actions, meaning that their work and effect can sometimes be transparent to the users (Figure 1).



Fig. 1. Automatically triggered workflow process.

- *Manual.* A manual workflow process requires explicit user intervention for its initiation (Figure 2).

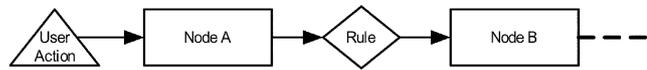


Fig. 2. Manually triggered workflow process.

- *Ad hoc workflow.* Unlike a predefined process, ad hoc workflow processes are not defined as part of the organisational modelling, but rather are created on demand by the user (Figure 3).

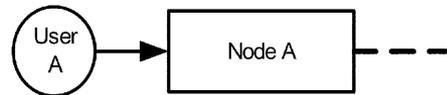


Fig. 3. Ad hoc workflow process.

4. The Workflow Meta-Model

The meta-model identifies the basic set of entities used in the workflow process definition. The top-level entities are as follows:

4.1. Nodes

A (workflow processing) node can be either implicit or explicit. An *implicit node* is a pointer to a task carried out by one (and only one) role. An example node is 'Review CVs', which is performed by the Personnel Administrator. Ultimately, the workflow engine will have to resolve the node to a specific user. An *explicit node* represents a specific user.

4.2. Rules

Rules are used to authorise (or not) the progression of a business document across the workflow path. Rules evaluate to true or false (carry on/do not carry on) and apply on either the fields of the business document under transfer, or the fields of one or more external business documents (Figure 4).

A business document instance can only participate in one workflow process at any one time.

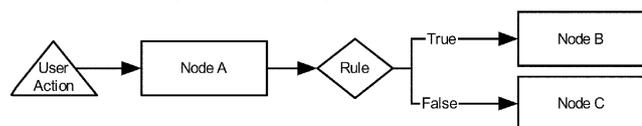


Fig. 4. Rules in a workflow process.

4.3. Workflow Assignment Rights

Workflow assignment rights are attached to each node and define the rights that the user has over the business document when it arrives. More specifically:

1. *Accept/decline*. When a user has ‘accept/decline’ rights over a business document at a specific node in a workflow process, they have the right not to accept the document. In such an instance, the user has the ability to break a process.
2. *Forward*. When a user has ‘forward’ rights over a business document at a specific node in a workflow process, they have the right to send (forward) the business document to another user (with the same role and under the same task).

Workflow assignment rights are defined as part of the workflow process definition.

4.4. Events and Exceptions

Events are pre-modelled incidents that happen within the context of our system and the organisation.

We categorise events as follows:

1. Creation of a new instance of a business document
2. Modifications to an existing business document
3. Deletion of a business document
4. Arrival or departure of a business document to or from a node
5. Change of a business documents state
6. Transformation of a business document (a business document is changed to another business document)
7. Start/end of a process
8. Scheduled event (timer)

A *workflow exception* is a special kind of event that generally represents issues that break the workflow process. Exceptions can be fired either by the fact that a user is not available to accept an assignment, the user has declined the assignment, or that the assignment has expired.

4.5. Business Document

A business document is an entity that encapsulates knowledge that is of value to an organisation. A business document encapsulates the following:

- Information (data)
 - Meaning of that information (meta-data)
 - Presentation information (layout)
 - Transformation information to other business documents
 - Workflow information
 - Links (relationship information to other business documents, files or URLs)
- } The combination of those two form what we call ‘knowledge’

Examples of business documents are an invoice, a purchase order, a contract, a memo and an email.

Information can be usefully divided into three different classes: *structured*, *unstructured* and *semi-structured*. XML documents, and databases, whether relational or otherwise, are examples of structured information sources. Examples of unstructured information include text files, word documents, emails and a large class of web pages. Examples of semi-structured information include web pages with known fields of contents.

Structured and semi-structured information can be modelled as XML documents and be part of the business document (BD) definition itself. Unstructured information, on the other hand, can be represented as a BD using links.

For example, if a map is the entity that we want to represent as a BD, we provide a high-level description of the map within the business document ‘map’ and we create a link of the actual map file (picture) with the business document.

4.6. Business Document State

Business document state represents the status of a business document during its life cycle and within a workflow process. Example states are: *Created*, *Proposed*, *Under Review*, *Approved*, *Pending*, *Transformed*, *Closed* and *Deployed* (Figure 5). BD states are defined as part of the workflow process definition. The user can define new states.

Three types of document are considered, namely (Figure 6):

1. Document instance
2. Process instance
3. Document template

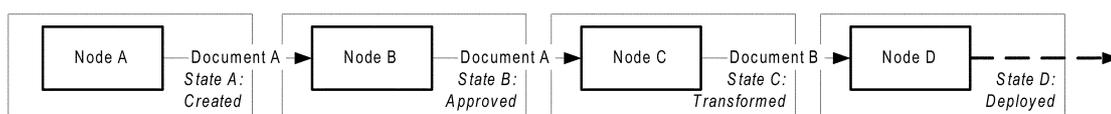


Fig. 5. Possible states of a business document.

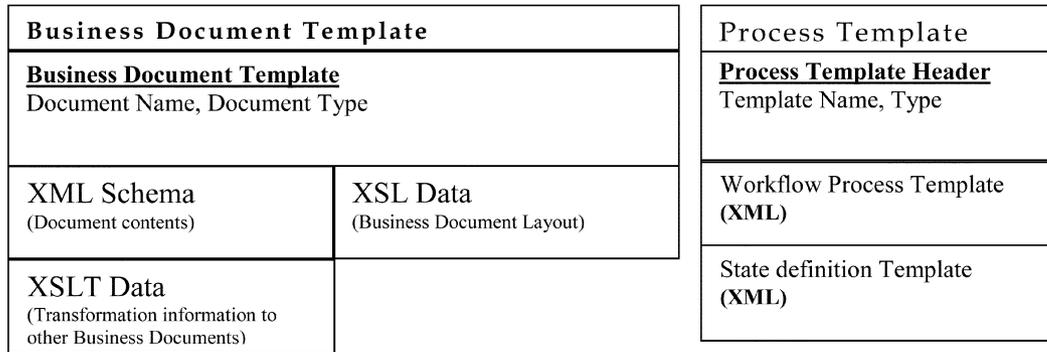


Fig. 6. Structure of process and business document templates.

Table 1. Business document template

```

<Business_document_class >
  <class_id::String >
  <class_name::String >
  <class_description::String >
</Business_document_class >
<Business_document_template >
<document_template_header >
<document_name::String >
  <document_description::String >
  <document_class_id::String >
</document_template_header >
<XML_Schema::XMLDocument >
<XSL_Schema::XMLDocument >
<XSLT_Data >
  <document_id::String >
  <XSLT_Data::XMLDocument >
</XSLT_Data >
<permissions_template >
</permissions_template >
</Business_document_template >

```

Table 2. Workflow process template

```

<Process_class >
<class_id::String >
<class_name::String >
  <class_description::String >
</Process_class >
<Workflow_process_template >
<process_template_header >
  <author::String >
  <Version::Long Integer >
  <creation_date::Date >
  <process_class_id::String >
  <process_description::String >
  <priority::Enum(High, Normal, Low) >
  <expected_duration::Time(Hrs) >
</process_template_header >
<workflow_path >
  <logical_node >
    <node_id::String >
    <organization_id::String >
    <business_unit_id::String >
    <task_id::String >
    <business_document_id::String >
    <document_state_id::String >
  </logical_node >
  <rule >
    <rule_id::String >
    <rule_name::String >
    <rule::String >
  </rule >
  <transition >
    <source_node_id::String >
    <rule_id::String >
    <destination_node_id::String >
  </transition >
</workflow_path >
<state_definition >
  <state_id::String >
  <state_name::String >
  <state_description::String >
</state_definition >
</Workflow_process_template >

```

These document types are defined as XML schemas in Tables 1, 2 and 3 respectively.

5. Case Study: Workflow Requirements Modelling in a Publishing Company

In this case study we apply the workflow concepts described in the previous section to model the workflow requirements of the Product Build Department of a real-life electronic publishing company. This case study provides examples of all the workflow types described in Section 3, i.e. internal and external workflow, administrative and production workflow etc. The Product Build Department receives the data from Editorial and then processes this data to produce the CD and the corresponding documentation (called InfoBase) for the website. During this phase there is testing involving

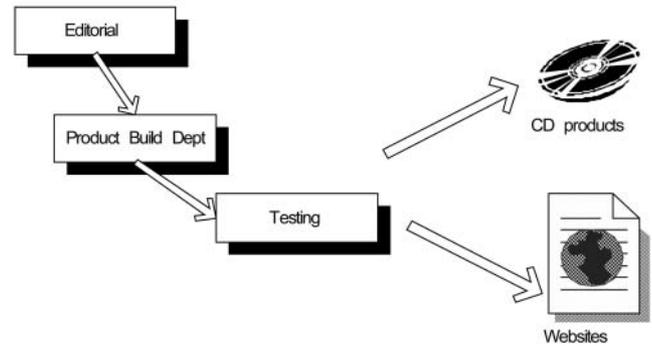
several other departments such as Helpdesk, Business Development and Editorial, and quite often the InfoBase has to be rebuilt.

Table 3. Business document instance

```

<Business_document_class> // e.g. Financial, Shipping, Banking
  <class_id::String>
  <class_name::String>
  <class_description::String>
</Business_document_class>
<Business_document_instance>
<document_header>
  <document_id::String>
  <document_name::String>
  <document_description::String>
  <document_class::String>
<document_revision::String>
<responsible>
  <role_id::String>
  <user_id::String>
</responsible>
</document_header>
<link>
  <link_id::String>
  <link_type::Enum(BD, File, Link)>
  <link_address::String>
</link>
<transport_header> // added by the workflow engine before
  <next_node_id::String> // it is sent to the transport engine.
  <address::String>
  <service_type_id::String> // e.g. SMTP, TCP/IP, FTP
</transport_header>
<permissions> // permissions string representing who
  // can and cannot view the object.
</permissions>
<XML_Schema::String>
<XSL_Schema::String>
<XSLT_Data::XMLDocument>
<state_definition> // definition of all the states this
  <state_id::String> // document can be in.
  <state_name::String>
  <state_description::String>
</state_definition>
// A business document only keeps the part of the process
// that is related to it. The whole process is depicted in
// the process template.
<workflow_path>
  <process_id::String>
  <logical_node>
    <node_id::String>
    <organization_id::String>
    <business_unit_id::String>
    <task_id::String>
    <state_id::String>
    <acceptance_rights::Enum(Accept, Decline, Forward)>
    <link_rights>
      <link_id::String>
      <access_rights::Enum(View, Edit)>
    </link_rights>
  </logical_node>
  <priority::Enum(High, Normal, Low)>
  <expiration::DateTime>
</workflow_path>
  <rule>
    <rule_id::String>
    <name::String>
    <rule::String>
  </rule>
<transition>
  <source_node_id::String>
  <rule_id::String>
  <destination_node_id::String>
</transition>
</workflow_path>
<workflow_history>
<process_id::String> // This holds the workflow
  <transformed_from_bd_id::String> // history of the BD in
  // the CURRENT process
<node>
  <node_id::String>
  <organization_id::String>
  <business_unit_id::String>
  <task_id::String>
  <state_id::String>
  <user_id::String>
</node>
<in_date::DateTime>
  <out_date::DateTime>
  <process_status::Enum(Completed, Broken, Suspended)>
</workflow_history>
</Business_document_instance>

```

**Fig. 7.** High-level view of workflow.

5.1 Roles and Tasks

The Team Leader manages the team and is responsible for minor decisions. The Team Leader can, for example, decide on whether to proceed with a particular change or not, and is also in contact with Editorial and Business Development Departments to get their approval for any changes made to the current release of the CD.

The external supplier, once they have processed the data, sends back the finished InfoBase. All that needs to be done is to verify the content and functionality and create an installation. Once that is done it is sent to the Helpdesk.

The Software Support Analyst plays a vital role in getting the CDs out. The build programs that are used are very difficult to maintain as these programs have been written in a variety of languages and for many of them the source code is not available. Also these programs had been written for a particular type of data and the data is constantly changing in terms of structure, so the build programs often crash.

The people from the Helpdesk are responsible for testing. They test the functionality and the installation of the CD. There are four different types of installation for the CDs, namely full hard drive installation, CD installation, network HD installation and the network CD installation.

The different titles that are dealt with by the department are assigned to the Product Build Technicians; hence it is they who are responsible for the smooth running of the process.

The Project Controller keeps track of all the titles that have to go under production and of the schedules that have to be respected for the data to be delivered on time. The editors are responsible for the data; they normally supply the data on a CD and a checklist form that the Product Build Technician has supplied. The editors ensure that the Product Build Department receives all

Table 4. Roles and tasks

Role	Tasks
Product Controller	Chases Editorial for data Prepares the schedules for the CD
Team Leader	Follows up of the progress of the build
Software Support Analyst	Supports the build programs Makes changes to the build programs
Product Build Technician/Assistant Helpdesk	Manages the build Tests the functionality of the CD Tests the installation of the CD
Editor	Produces CD data
Technical Editor	Makes changes to the CD data
Business Development Person	Verifies the quality of the CD Signs off of the CD
External Supplier	Processes the data for CD Processes data for the Websites

up-to-date data including the latest graphics. If any data is missing the technicians will go back to the editor responsible for the title.

If the data need to be modified, this task is performed by the Technical Editor.

In Business Development, a person will sign off the CD, checking the quality of the product title and authorising the CDs to go for replication.

The Product Build Department is made up of two teams, with two Team Leaders, i.e. Product Coordinator and Head of Department. Each team is made up of Product Build Technicians and/or Product Build Assistants and one Software Support Analyst, with the Team Leader reporting to the Head of Department.

Table 4 provides a list of all the roles and tasks that are involved in the production of the CDs and online content.

5.2 Description of the Workflow

The Product Build Department deals with the various kinds of CD builds. The workflow described below gives an overview of the typical builds that take place in the department. We shall focus on the processes of producing the CD and of going live.

There are two main types of CD builds in the department: the in-house and the out-of-house. They are different in that with out-of-house builds all the processing is done outside by the supplier and the supplier sends back the final InfoBase once the data has been processed.

5.2.1. In-House Builds

The process for in-house builds is fairly straightforward. However, if things go wrong during the build process it can become far more complex than the out-of-house process. The process consists of the four stages described below.

Stage 1: Data supply. Each product has a Content Manager in the Editorial Department, who is responsible for putting together the appropriate data set, which is then passed to the Product Build Department. The data is then put through quality checks. This is done by the Data Conversion Unit (DCU); if all is OK, DCU will send to the Product Build Department a CD with the data together with an inventory of contents from the Content Manager. If there are any problems these are subject to a Data Change Request (DCR), and the Content Manager is responsible for ensuring these

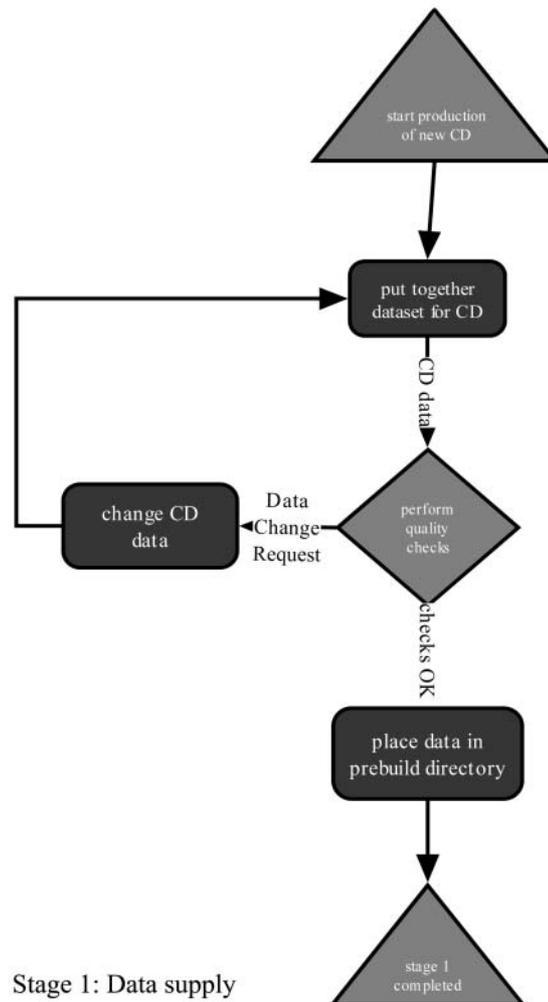


Fig. 8. Workflow model for data supply.

changes are made. These changes often have to be made after the build is done as it is very difficult to verify the content initially. However, if something goes wrong in the build, the Software Support Analyst will investigate and if the problem is with the data a DCR form will be completed and sent to the Content Manager. Once the Product Build Department has been provided with a complete set of data that passes the QC checks (on a CD from the Content Manager) the data is placed in the appropriate directories within a pre-set-up build directory on the appropriate machine. It is the task of the Product Build Technician to set up and kick off the build. The graphical model of Stage 1 workflow is shown in Fig. 8.

Stage 2: Batch files and building. Each product has a 'batch file'. This is a file that contains 'call-outs' to start different programs within the build process. The batch file will need to be checked to ensure that it is set to start at the right program. During the build process the programs can be paused and restarted at any point so

when starting from scratch you need to check that in the list at the top of the batch file. In an ideal world, once the build has started it should go through ideally without any problems; however, things do not go that smoothly at all. Very often the build crashes for several reasons such as incorrect data or hardware problems such as insufficient memory etc.

Stage 3: Product check. As mentioned earlier, if the product build is OK, the InfoBase must then be checked. Also the contents are checked against the proofs and filing instructions to ensure the correct set of data is in the InfoBase.

Stage 4: Binding and handover. Once the InfoBase is ready the Product Build Technician will create an installation and test the CD; in some cases another person can do the testing. The Team Leader then signs the necessary forms so that the CD can be sent to a different department for testing.

Stages 2–4 are shown in the workflow diagram of Fig. 9.

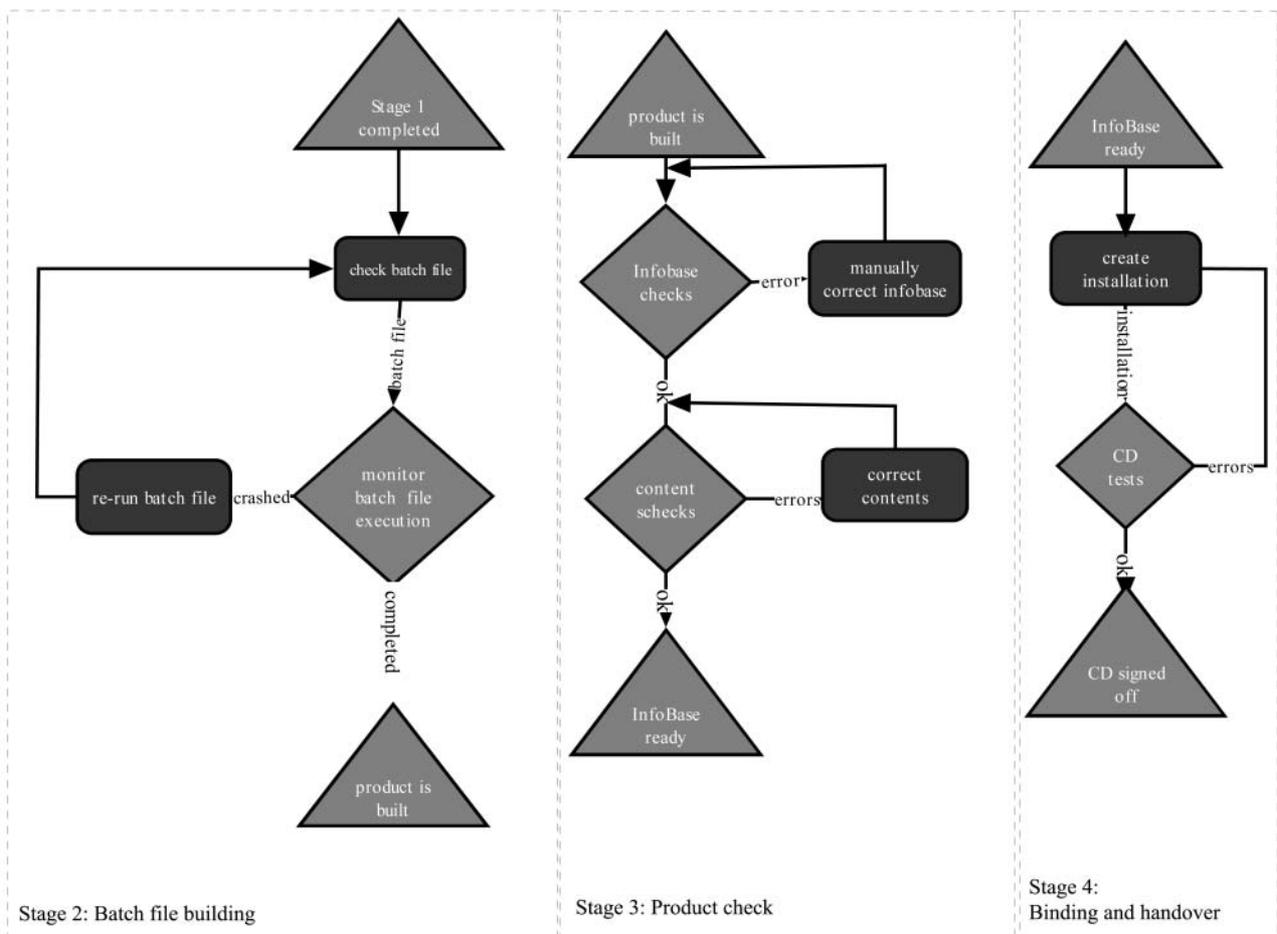


Fig. 9. Stages 2–4 of in-house builds.

5.2.2. Out-of-House Builds

Stage 1: Pre-build. For the majority of these titles the Product Build Technician does not see the data. Data is sent straight to the external supplier by the Product Controller and/or Technician. The data is parsed and verified against the checklist before being sent to the supplier. This is done to so that the supplier receives all the data in the appropriate format.

Stage 2: Receiving an InfoBase. Once the data has been sent to the supplier they take up to two weeks (depending on the title) to return an InfoBase to the department. The Product Build Technician then has to check it thoroughly against the proofs.

Stage 3: Binding and handover. Binding is an important task that needs to be done before the company can deliver an InfoBase; it has to be bound and protected by a password. Binding makes the InfoBase proprietary to the company and also keeps a record of the royalties that have to be paid. Once the binding is done, the technician will create an installation, check it and then send it to the other departments for approval, similarly to the out-of-house titles.

The out-of-house workflow process is modelled in a similar manner to the in-house one and is therefore not shown.

6. Conclusions

To provide a method to describe workflows, a workflow process definition meta-data model has been described in this paper. This meta-data model identifies commonly used entities within a process definition such as event, task/role and business document. A variety of attributes describe the characteristics of this set of entities encoded as XML schemas.

We are currently working on a workflow editor that is capable of defining a workflow process using the notation described in this paper and from it automatically generating a WfMC-compatible schema. Once completed the editor will be capable of generating an XPDL version of the process such as the one shown in the Appendix. This output is essentially a common interchange format, which supports the transfer of workflow process definitions between separate products.

Further work could extend the XPDL language to allow the expression of more elaborate workflow constructs such as complex rules, although the impact of such extensions on the portability of the models needs to be investigated.

References

1. Pelz-Sharpe A, Ashenden A. E-process: workflow for the e-business. Ovum report, Ovum Ltd, London, 2001
2. Workflow Management Coalition (WfMC). (2001), Workflow standard-workflow process definition interface. XML Process Definition Language. Document no WFMC-TC-1025, Florida, USA, 2001
3. Workflow Management Coalition (WfMC). The workflow reference model. WfMC Document TC00-1003, Florida, USA, November 1994
4. Adam NR, Atluri V, Huang WK. Modelling and analysis of workflows using Petri nets. *J Intell Inform Syst* 1998;10(2):131–158
5. Wagner C, Chaudhury A, Raghav RH, Sanders GL. (1995), A workflow approach to information systems development. In: 1995 inaugural Americas conference, Association for Information Systems, Pittsburgh, 1995
6. Hruby P. Structuring specification of business systems with UML (with an emphasis on workflow management systems). In: OOPSLA '98 business object workshop IV, 1998
7. OMG. Workflow management facility specification. OMG document bom/98-03-01, MA, USA, 1998
8. Schmidt M. Building workflow business objects. In: OOPSLA '98 business object workshop IV, Vancouver, Canada, 1998

APPENDIX: The Workflow process in XPDL

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!-
edited with XML Spy v4.3 U (http://www.xmlspy.com) (Home)
->
<?xml-stylesheet type="text/xsl" href=""?>
≈ <Package Id="Workflow" Name="Product Build Department
Package">
≈ <PackageHeader>
<XPDLVersion>0.02</XPDLVersion>
<Vendor>S&M:workflow:1.0</Vendor>
<Created>February 2002</Created>
<Description>Workflow for the Product Build Department
</Description>
<Documentation>help.xml</Documentation>
<PriorityUnit />
</PackageHeader>
≈ <RedefinableHeader PublicationStatus="UNDER_TEST">
<Author>WM</Author>
<Version>1.0</Version>
<Codepage />
<Countrykey>44</Countrykey>
≈ <Responsibles>
<Responsible>Product Build Technician</Responsible>
</Responsibles>
</RedefinableHeader>
<ConformanceClass GraphConformance="NON_BLOCKED" />
≈ <Participants>
≈ <Participant Id="Product_Build_Technician" Name=
"Product Build Technician">
<ParticipantType Type="ROLE" />
<Description>Person responsible for the build</Description>
</Participant>
≈ <Participant Id="Software_Support_Analyst" Name="Software
Support Analyst">
<ParticipantType Type="ROLE" />
<Description>Person giving support to the build
</Description>
```

```

</Participant>
≈ <Participant Id="Team_Leader" Name="Team Leader">
  <ParticipantType Type="ROLE" />
  <Description>Person Responsible for the product
</Description>
</Participant>
≈ <Participant Id="Editor" Name="Editor">
  <ParticipantType Type="ROLE" />
  <Description>Person responsible for the content
</Description>
</Participant>
≈ <Participant Id="Business_Analyst" Name="Business Analyst">
  <ParticipantType Type="ROLE" />
  <Description>Person responsible for the whole product
</Description>
</Participant>
≈ <Participant Id="Helpdesk" Name="Helpdesk">
  <ParticipantType Type="ORGANIZATIONAL_UNIT" />
  <Description>Testing the CDs</Description>
</Participant>
≈ <Participant Id="Supplier" Name="Supplier">
  <ParticipantType Type="ORGANIZATIONAL_UNIT" />
  <Description>External supplier that process the data
</Description>
</Participant>
≈ <Participant Id="Technical_Editor" Name="Technical
  Editor">
  <ParticipantType Type="ROLE" />
  <Description>Person responsible for the content and that
  makes changes to the data</Description>
</Participant>
≈ <Participant Id="Production" Name="Production">
  <ParticipantType Type="ORGANIZATIONAL_UNIT" />
  <Description>Department for posting the CDs sending the
  CDs for replication</Description>
</Participant>
</Participants>
≈ <Applications>
  ≈ <Application Id="Build_programs" Name="Build programs">
    <Description>Suite of programs used to build the CDs and to
    process the infobase prior to go live</Description>
  </Application>
  ≈ <Application Id="Folio_Views" Name="Folio Views">
    <Description>Software used to view the infobase
    </Description>
  </Application>
  ≈ <Application Id="InstallShield" Name="InstallShield
    Programs">
    <Description>Software used to distribute our Cds
    </Description>
  </Application>
  ≈ <Application Id="Folio_WorkBench" Name="Folio
    WorkBench">
    <Description>Software used to generate the infobase
    </Description>
  </Application>
  ≈ <Application Id="Internet_Browser" Name="Internet
    Browser">
    <Description>Browser to view the online products
    </Description>
  </Application>
  ≈ <Application Id="Folio_NextPage" Name="Folio Next Page">
    <Description>Software use to create a document collection
    file</Description>
  </Application>
  </Applications>
  ≈ <Application>
    </Application>
  ≈ <Application Id="FileTransfer_Application" Name="File
    Transfer Application">
    <Description>Software used to transfer files to and from the
    supplier</Description>
  </Application>
  ≈ <Application Id="Folio_Publisher" Name="Folio Publisher">
    <Description>Software used to bind the infobase
    </Description>
  </Application>
</Applications>
≈ <DataFields>
  ≈ <DataField Id="FileNames" Name="FileNames">
    ≈ <DataType>
      <BasicType Type="STRING" />
    </DataType>
    <Length>1024</Length>
    <Description>All the filenames to be used during the build
    </Description>
  </DataField>
  ≈ <DataField Id="Number_Files" Name="Number of files
    required to launch the build">
    ≈ <DataType>
      <BasicType Type="INTEGER" />
    </DataType>
  </DataField>
  ≈ <DataField Id="Build_OK" Name="Build OK">
    ≈ <DataType>
      <PlainType Type="BOOLEAN" />
    </DataType>
  </DataField>
  ≈ <DataField Id="Build_Start_Date" Name="Build Start Date">
    ≈ <DataType>
      <BasicType Type="DATETIME" />
    </DataType>
  </DataField>
  ≈ <DataField Id="Build_End_Date" Name="Build End Date">
    ≈ <DataType>
      <BasicType Type="DATETIME" />
    </DataType>
  </DataField>
  ≈ <DataField Id="Inventory_Ok" Name="Inventory Ok">
    ≈ <DataType>
      <PlainType Type="BOOLEAN" />
    </DataType>
  </DataField>
  ≈ <DataField Id="CD_OK" Name="CD OK">
    ≈ <DataType>
      <PlainType Type="BOOLEAN" />
    </DataType>
  </DataField>
  ≈ <DataField Id="Site_OK" Name="Site OK">
    ≈ <DataType>
      <PlainType Type="BOOLEAN" />
    </DataType>
  </DataField>
</DataFields>
≈ <WorkflowProcesses>
  ≈ <WorkflowProcess Id="Product_Build_Department"
    Name="Workflow for the Product Build Department">
    ≈ <ProcessHeader DurationUnit="D">
      <Created>February 2002</Created>
      <Description>Process of building CDs and putting content
      online</Description>
    </ProcessHeader>
  </WorkflowProcess>

```

```

    <Priority />
    <Limit>0</Limit>
    <ValidFrom />
    <ValidTo />
    ≈ <TimeEstimation>
      <WaitingTime>1D5/WaitingTime>
      <WorkingTime>7D5/WorkingTime>
      <Duration>D5/Duration>
    </TimeEstimation>
  </ProcessHeader>
  ≈ <RedefinableHeader PublicationStatus="UNDER_TEST">
    <Author>WM</Author>
    <Version>1.0</Version>
    <Codepage />
    <Countrykey>United Kingdom</Countrykey>
    ≈ <Responsibles>
      <Responsible>Product Build Technician</Responsible>
    </Responsibles>
  </RedefinableHeader>
  ≈ <Activities>
    ≈ <Activity Id="Data_Handover" Name="Data Handover">
      <Description>Editor Handing the Data to the Product Build
        Technician</Description>
      <Limit>0.5D</Limit>
    ≈ <Implementation>
      <No />
    </Implementation>
    <Performer>Editor</Performer>
    ≈ <StartMode>
      <Manual />
    </StartMode>
    ≈ <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
    ≈ <Activity Id="Inventory_Check" Name="Inventory Check">
      <Description>Check whether all the files have been submitted
        and that the filenames are correct</Description>
    ≈ <Implementation>
      ≈ <Loop Kind="REPEAT_UNTIL">
        <Condition>Number of files and filenames OK</Condition>
      </Loop>
    </Implementation>
    <Performer>Product Build Technician</Performer>
    ≈ <StartMode>
      <Manual />
    </StartMode>
    ≈ <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
    ≈ <TransitionRestrictions>
      ≈ <TransitionRestriction>
        ≈ <Split Type="XOR">
          ≈ <TransitionRefs>
            <TransitionRef Id="t_11" />
            <TransitionRef Id="t_12" />
            <TransitionRef Id="t_13" />
          </TransitionRefs>
        </Split>
      </TransitionRestriction>
    </TransitionRestrictions>
  </Activity>
  ≈ <Activity Id="Re-Supply_Data" Name="Re-supply Data">

```

```

    <Description>Editor make the necessary changes requested by
      Product Build Technician</Description>
    ≈ <Implementation>
      <No />
    </Implementation>
    <Performer>Editor</Performer>
    ≈ <StartMode>
      <Manual />
    </StartMode>
    ≈ <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
  ≈ <Activity Id="Data_to_Supplier" Name="Data to supplier">
    <Description>Send the data to the supplier to be processed
      </Description>
    ≈ <Implementation>
      <No />
    </Implementation>
    <Performer>Product Build Technician</Performer>
    ≈ <StartMode>
      <Manual />
    </StartMode>
    ≈ <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
  ≈ <Activity Id="Supplier_build" Name="Supplier build">
    <Description>The build process is run at supplier
      </Description>
    ≈ <Implementation>
      <No />
    </Implementation>
    <Performer>Supplier</Performer>
    ≈ <StartMode>
      <Manual />
    </StartMode>
    ≈ <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
  ≈ <Activity Id="Return_Infobase" Name="Return Infobase">
    <Description>The processed and final infobase is returned to
      S&M</Description>
    ≈ <Implementation>
      <No />
    </Implementation>
    <Performer>Supplier</Performer>
    ≈ <StartMode>
      <Manual />
    </StartMode>
    ≈ <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
  ≈ <Activity Id="Build_Setup" Name="Build Setup">
    <Description>Setting up the build</Description>
    ≈ <Implementation>
      ≈ <Loop Kind="REPEAT_UNTIL">
        <Condition>Infobase OK</Condition>
      </Loop>
    </Implementation>
    <Performer>Product Build Technician</Performer>
    ≈ <StartMode>

```

```

    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Route1" Name="Route1">
  <Route />
  <TransitionRestrictions />
</Activity>
≈ <Activity Id="Run_Build" Name="Build run">
  <Description> Running the build programs </Description>
  ≈ <Implementation>
    ≈ <Tool Type="APPLICATION" Id="Build_programs">
      <Description> Suite of programs used to convert the original
        data to an infobase </Description>
    </Tool>
    ≈ <Tool Id="Folio_WorkBench" Type="APPLICATION">
      <Description> Application use to create the nfo
      </Description>
    </Tool>
    </Implementation>
    <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Automatic />
  </FinishMode>
</Activity>
≈ <Activity Id="Check_build" Name="Checking the build">
  <Description> Checking the infobase </Description>
  ≈ <Implementation>
    <No />
  </Implementation>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Automatic />
  </FinishMode>
</Activity>
≈ <Activity Id="Change_Data" Name="Change data">
  <Description> Modifying the data </Description>
  ≈ <Implementation>
    <No />
  </Implementation>
  <Performer> Technical Editor </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Modify_programs" Name="Modify programs">
  <Description> Modify the build programs or batch files
  </Description>
  ≈ <Implementation>
    <No />
  </Implementation>
  <Performer> Software Support Analyst </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Manual_correction" Name="Manual Correction">
  <Description> Manual correction on the infobase </Description>
  ≈ <Implementation>
    <Tool Id="Folio_Views" Type="PROCEDURE">
      <Description> Application software used to open the infobase
        and make the manual correction </Description>
    </Tool>
    </Implementation>
    <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Bind_Infobase" Name="Bind Infobase">
  <Description> Binding the Infobase </Description>
  ≈ <Implementation>
    ≈ <Tool Id="Folio_Publisher" Type="APPLICATION">
      <Description> Application package used to bind the infobase
      </Description>
    </Tool>
    </Implementation>
    <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Automatic />
  </FinishMode>
</Activity>
≈ <Activity Id="Create_Install" Name="Create Install">
  <Description> Create and install program for the infobase
  </Description>
  ≈ <Implementation>
    ≈ <Tool Id="InstallShield" Type="APPLICATION">
      <Description> Application Package use to create the install for
        the CDs </Description>
    </Tool>
    </Implementation>
    <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Automatic />
  </FinishMode>
</Activity>
≈ <Activity Id="Goldmaster" Name="Goldmaster">
  <Description> Sending the goldmaster to production
  </Description>
  ≈ <Implementation>
    <No />
  </Implementation>
  <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>

```

```

≅ <FinishMode>
  <Manual />
</FinishMode>
≅ <TransitionRestrictions>
  ≅ <TransitionRestriction>
    ≅ <Split Type="AND">
      ≅ <TransitionRefs>
        <TransitionRef Id="t_17" />
        <TransitionRef Id="t_18" />
        <TransitionRef Id="t_19" />
      </TransitionRefs>
    </Split>
  </TransitionRestriction>
</TransitionRestrictions>
</Activity>
≅ <Activity Id="Helpdesk_Test" Name="HelpDesk Test">
  <Description>HelpDesk testing the goldmaster</Description>
  ≅ <Implementation>
    ≅ <Loop Kind="REPEAT_UNTIL">
      <Condition>CD_OK</Condition>
    </Loop>
  </Implementation>
  <Performer>Helpdesk</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Editorial_Test" Name="Editorial Test">
  <Description>Editorial testing the content</Description>
  ≅ <Implementation>
    <No />
  </Implementation>
  <Performer>Editorial</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Bus_Dev_Test" Name="Business Development
  Test">
  <Description>Business testing quality of the Cd and
    approving</Description>
  ≅ <Implementation>
    <No />
  </Implementation>
  <Performer>Business Analyst</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Route2" Name="Route2">
  <Route />
  ≅ <TransitionRestrictions>
    ≅ <TransitionRestriction>
      <Join Type="AND" />
    </TransitionRestriction>
  </TransitionRestrictions>
  ≅ <TransitionRefs>
    <TransitionRef Id="t_23" />
    <TransitionRef Id="t_24" />
    <TransitionRef Id="t_25" />
  </TransitionRefs>
  </Split>
  </TransitionRestriction>
</TransitionRestrictions>
</Activity>
≅ <Activity Id="Replicate_CD" Name="Replicate Cd">
  <Description>The CD is replicated</Description>
  ≅ <Implementation>
    <No />
  </Implementation>
  <Performer>Production</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Final_Check" Name="Final Check">
  <Description>HelpDesk makes a final check on the Cd that
    will be posted</Description>
  ≅ <Implementation>
    <No />
  </Implementation>
  <Performer>Helpdesk</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Post_CD" Name="Post CD">
  <Description>Post the Cds to the customers</Description>
  ≅ <Implementation>
    <No />
  </Implementation>
  <Performer>Production</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Backup" Name="Backup">
  <Description>Backup of the programs, infobase and data
    </Description>
  ≅ <Implementation>
    <No />
  </Implementation>
  <Performer>Product Build Technician</Performer>
  ≅ <StartMode>
    <Manual />
  </StartMode>
  ≅ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≅ <Activity Id="Process_Infobase" Name="Process Infobase">

```

```

    <Description> Process the Infobase for the upload
  </Description>
  ≈ <Implementation>
    <No />
  </Implementation>
  <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Upload_Document" Name="Upload document">
  <Description> Upload the document collection </Description>
  ≈ <Implementation>
    ≈ <Tool Id="FileTransfer_Application"
      Type="APPLICATION">
      <Description> Application package use to transfer the file on
        the server to go live </Description>
    </Tool>
  </Implementation>
  <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Check_site1" Name="Check site">
  <Description> Check the look content and functionality of
    the website </Description>
  ≈ <Implementation>
    ≈ <Tool Id="Internet_Browser"
      Type="PROCEDURE">>
    <Description> View the test site before going live </Description>
  </Tool>
  </Implementation>
  <Performer> Product Build Technician </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
≈ <Activity Id="Go_live" Name="Go live">
  <Description> Go live with the website </Description>
  ≈ <Implementation>
    <No />
  </Implementation>
  <Performer> Team Leader </Performer>
  ≈ <StartMode>
    <Manual />
  </StartMode>
  ≈ <FinishMode>
    <Manual />
  </FinishMode>
</Activity>
</Activities>
≈ <Transitions>
  ≈ <Transition Id="t_1" From="Data_Handover"
    To="Inventory_Check" Loop="NOLOOP">
    Name="Data_Handover to Inventory_Check">
    <Description> Transition from the data handover to
      inventory check </Description>
  </Transition>
  ≈ <Transition Id="t_2" From="Inventory_Check1"
    To="Re-Supply_Data" Loop="FROMLOOP">
    Name="Re-supply">
    <Condition Type="CONDITION"> Files supplied OK
  </Condition>
  </Transition>
  <Transition Id="t_3" From="Inventory_Check"
    To="Data_to_Supplier" Loop="NOLOOP" />
  <Transition Id="t_4" From="Inventory_Check"
    To="Build_Setup" Loop="NOLOOP" />
  <Transition Id="t_5" From="Data_to_Supplier"
    To="Supplier_build" Loop="NOLOOP" />
  <Transition Id="t_6" From="Supplier_build"
    To="Return_Infobase" Loop="NOLOOP" />
  <Transition Id="t_7" From="Build_Setup"
    To="Run_Build" Loop="NOLOOP" />
  <Transition Id="t_8" From="Return_Infobase"
    To="Route1" Loop="NOLOOP" />
  ≈ <Transition Id="t_9" From="Run_Build"
    To="Route1" Loop="FROMLOOP">
    <Condition Type="CONDITION"> Infobase OK
  </Condition>
  </Transition>
  <Transition Id="t_10" From="Route1" To="Check_Build" />
  ≈ <Transition Id="t_11" From="Check_build"
    To="Modify_programs" Loop="FROMLOOP">
    <Condition Type="CONDITION"> Infobase OK
  </Condition>
  </Transition>
  <Transition Id="t_12" From="Check_build"
    To="Change_Data" Loop="FROMLOOP">
    <Condition Type="CONDITION"> Infobase OK
  </Condition>
  </Transition>
  <Transition Id="t_13" From="Check_build"
    To="Manual_correction" Loop="NOLOOP" />
  <Transition Id="t_14" From="Manual_correction"
    To="Bind_Infobase" Loop="NOLOOP" />
  <Transition Id="t_15" From="Bind_Infobase"
    To="Create_Install" Loop="NOLOOP" />
  <Transition Id="t_16" From="Create_Install"
    To="Goldmaster" Loop="NOLOOP" />
  ≈ <Transition Id="t_17" From="Goldmaster"
    To="Helpdesk_Test" Loop="FROMLOOP">
    <Condition Type="CONDITION"> CD OK
  </Condition>
  </Transition>
  ≈ <Transition Id="t_18" From="Goldmaster"
    To="Editorial_Test" Loop="FROMLOOP">
    <Condition Type="CONDITION"> CD OK
  </Condition>
  </Transition>
  ≈ <Transition Id="t_19" From="Goldmaster" To="Bus_Dev_
    Test" Loop="FROMLOOP">
    <Condition Type="CONDITION"> CD OK
  </Condition>
  </Transition>

```

```

<Transition Id="t_20" From="Helpdesk_Test" To="Route2"
  Loop="NOLOOP" />
<Transition Id="t_21" From="Editorial_Test" To="Route2"
  Loop="NOLOOP" />
<Transition Id="t_22" From="Bus_Dev_Test" To="Route2"
  Loop="NOLOOP" />
<Transition Id="t_23" From="Route2" To="Replicate_CD"
  Loop="NOLOOP" />
<Transition Id="t_24" From="Route2" To="Backup"
  Loop="NOLOOP" />
<Transition Id="t_25" From="Route2" To="Process_Infobase"
  Loop="NOLOOP" />
<Transition Id="t_26" From="Replicate_CD"
  To="Final_Check" Loop="NOLOOP" />

```

```

<Transition Id="t_27" From="Final_Check"
  To="Post_CD" Loop="NOLOOP" />
≅ <Transition Id="t_28" From="Process_Infobase"
  To="Upload_Document" Loop="NOLOOP">
  <Condition Type="CONDITION"> Site OK
  </Condition>
</Transition>
<Transition Id="t_29" From="Upload_Document"
  To="Check_site" Loop="NOLOOP" />
<Transition Id="t_30" From="Check_site" To="Go_live"
  Loop="NOLOOP" />
</Transitions>
</WorkflowProcess>
</WorkflowProcesses>
</Package>

```