

YAWL4Film: Film Production Process Automation Documentation

Chun Ouyang
c.ouyang@qut.edu.au

BPM Research Group
Faculty of Science and Technology
Queensland University of Technology (QUT)
GPO Box 2434, Brisbane QLD 4001, Australia

May 25, 2010

1 Introduction

This document describes YAWL4Film, a case study on applying YAWL to the automation of film production processes. The term “film production process” focuses on the chain of information processing accompanied with the shooting activities during a film production. The case study was undertaken in close collaboration with the Australian Film Television and Radio School (AFTRS). Through extensive discussions with domain experts from AFTRS, we were able to collect user requirements regarding film production process automation as well as to obtain and clarify domain-specific knowledge for filmmaking. The case study provides a demonstration of YAWL’s capabilities in modelling process with sophisticated data requirements and to support integration into a traditional workflow system the customised forms with professional look-and-feel.

2 What is a Film Production Process

A film production process covers the entire period of the actual shooting of a motion picture, which can be as short as a day or last for several years (e.g. a very large production). The shooting procedure is carried out on a daily basis. During each day, a number of activities such as acting, camera and sound recording are performed in studio or on location. While shooting is taking place, designated onset crew collect the information associated with each of these activities via corresponding production forms. Here are some of the examples. The *continuity* person (also known as the *script supervisor*) is in charge of the *continuity log* which records continuity details of each scene shot (e.g. the timing, the lighting condition, the takes of director’s choice), and after the shooting is wrapped up he/she also needs to fill the *continuity daily* which summarises of all the scenes shot on the day. The *camera assistant* is responsible for filling the *camera sheet(s)*¹ with the photographic details of each shot (e.g. the counter reading and the film length of each take). The sound recordist will complete the *sound sheet(s)* of all dialogue or effects recorded on set. The *second assistant director* (2nd AD) maintains a timesheet-like document called the *2nd AD report* for logging the data related to the working hours of the crew and cast, exceptions like delays and accidents, etc. All the above information is later gathered and collated to generate the *daily progress report* (DPR) at the *production office*. A DPR is a summary of the shooting information for each day and serves the purpose of keeping track of a production’s progress and expenses.

In addition to the generation of DPR, the production office monitors requirements for the subsequent shooting days and communicates frequently with onset crew to work out and finalise the *call sheet* for the next day. A call sheet is the one stop form for all of a particular shooting day’s logistics and necessities, and is usually issued to all cast and crew one day in advance. It takes as input the *shooting schedule* which is a project plan for the entire production period, and elaborates the daily schedule of scene shooting by adding detailed timings for each member of the production crew and cast as well as specific costume, location, make-up, prop, transport requirements and so on. Since the schedule of scenes often changes, a daily call sheet usually needs to be updated several times before it can be finalised.

Table 1 summarises the list of production documents aforementioned. All of them are created during the production except the shooting schedule which is created in the pre-production but is often revised and updated during the production.

¹Note that each camera (or sound) sheet is created for one camera (or sound) roll used on a day.

Place	Document Name	Person in Charge	Input Document
on set	continuity log continuity daily camera sheet sound sheet 2nd AD report	continuity continuity camera assistant sound recordist 2nd AD	- continuity log - - call sheet
production office	DPR	production coordinator (edit) production manager (review)	continuity daily camera sheet sound sheet 2nd AD report previous day's DPR
	call sheet	production coordinator (edit) production manager (review)	shooting schedule
(from pre-production)	shooting schedule	1st AD	cast list crew list location notes

Table 1: List of documents created/used during a film production.

3 The Film Production Process Model

3.1 Control-flow Perspective

Figure 1 depicts the YAWL model of a film production process. Tasks are represented as rectangles that may have an icon indicating whether they are manual or automatic. A task without an icon is an “empty” task that appears only for routing purposes. An instance of the process model starts with the collection of specific documents (i.e. cast list, crew list, location notes, and shooting schedule) generated during the pre-production phase. Next, the shooting starts and is carried out on a daily basis.

Each day tasks are performed along two main parallel subprocesses. One subprocess starts with task *Begin Call Sheet* and ends with task *Finish Call Sheet* in Fig. 1. It focuses on the production of a call sheet for the next day. The call sheet is created automatically from the shooting schedule and then updated with certain information that is not included in the schedule, such as individual cast and crew call times. The schedule of scenes for the subsequent shooting days often changes due to various reasons such as delay in the shooting of scheduled scenes, unexpected weather change, absence of designated cast and crew members, break-down of onset equipment, etc. Once the schedule has been revised, the call sheet can be re-created accordingly. In addition, the call sheet specific information may also need update during the production. In the subprocess of call-sheet production, the modelling of two deferred choices – one between task *Create Call Sheet* and task *Revise Shooting Schedule*, the other between task *Update Call Sheet* and task *Revise Shooting Schedule* – makes it possible to revise shooting schedule at any time before the call sheet is finalised, and the modelling of loops around these three tasks allows multiple executions of each of the tasks. Task *Update Call Sheet* has an XOR-split which determines whether to continue updating or to finalise the call sheet based on the user input collected by the task. The call sheet may go through any number of revisions and once finalised will be distributed to every cast and crew. Given that a call sheet is always produced one day in advance, no call sheet is generated on the last shooting day. This is captured by the

conditional outgoing branch of task *Begin Call Sheet* that is directly connected with task *Finish Call Sheet* thus bypassing the actual production of a call sheet on the last day.

The other subprocess starts with task *Kick off On-set* and ends with task *Distribute DPR* in Fig. 1. It specifies the sequence of onset shooting activities and supports the production of a DPR. Task *Kick off On-set* is an AND-join whose incoming flow diverges into four parallel outgoing branches to capture individual shooting activities. On each branch tasks are executed to record information of each shooting activity into the corresponding onset documents (i.e. continuity log and continuity daily, sound sheet, camera sheet, and 2nd AD Report). Usually, the production manager or 1st AD need to check the status of the shooting continuity at certain point of the day, e.g. during a meal break. In the process model, the loop containing a sequence of task *Fill Continuity Log* followed by task *Review Continuity Log*, allows the *continuity* person to submit a work-in-progress version of the continuity log for review, and then to resume the work after the review. In parallel, there may be multiple sound or camera sheets to be filled in during a day, which are captured by the self-loop of task *Fill Sound Sheet* or that of task *Fill Camera Sheet*. Next, upon completion of the above onset documents, a DPR is automatically generated and passed onto the Production Manager for review. After the review is finished, the DPR is circulated to certain crew members such as Producer and Executive Producer.

It is interesting to see how the OR-join associated with task *End a Day* behaves. Before the first shooting day starts, an instance of the call sheet branch is executed for producing the first day's call sheet. Since it is the only active incoming branch to task *End a Day*, the task will be performed once the call sheet has completed, without waiting for the completion of a DPR. In this case, the OR-join behaves like an XOR-join. On the other hand, if both call sheet and DPR branches are active (which is the case for the rest of the shooting days) the OR-join behaves like an AND-join. Finally, task *End a Day* has an XOR-split which determines whether it is the last shooting day. If so the entire process ends, otherwise the flow loops back to task *Start a Day* which initiates the shooting on the next day.

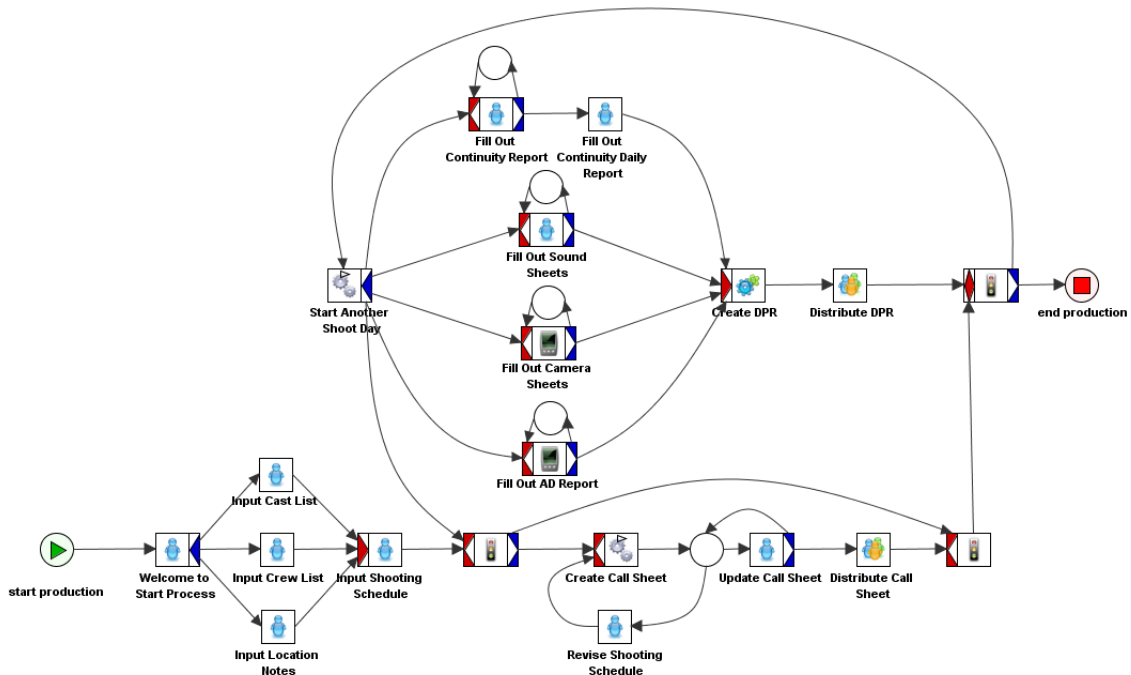


Figure 1: A film production process model in YAWL.

3.2 Data Definition

A film production is a data-intensive process. The YAWL model in Fig. 1 handles all the eleven production and pre-production documents listed in Table 1. To capture the information on these documents, a list of complex data types are defined using XML schema. For example, Fig. 2 sketches the data types used to specify the information on a call sheet. The *GeneralInfoType* captures the common information that is shown on each document. The *CallSheetInfoType* is the main data type that offers a structure for comprising the rest of the information on a call sheet. It is build upon a number of smaller data types which are defined to capture a multiple occurrence data structure (e.g. *SetReqInfoType*) or for reuse purposes (e.g. *SceneInfoType*). All these data types are valid and (re)usable within the entire process model.

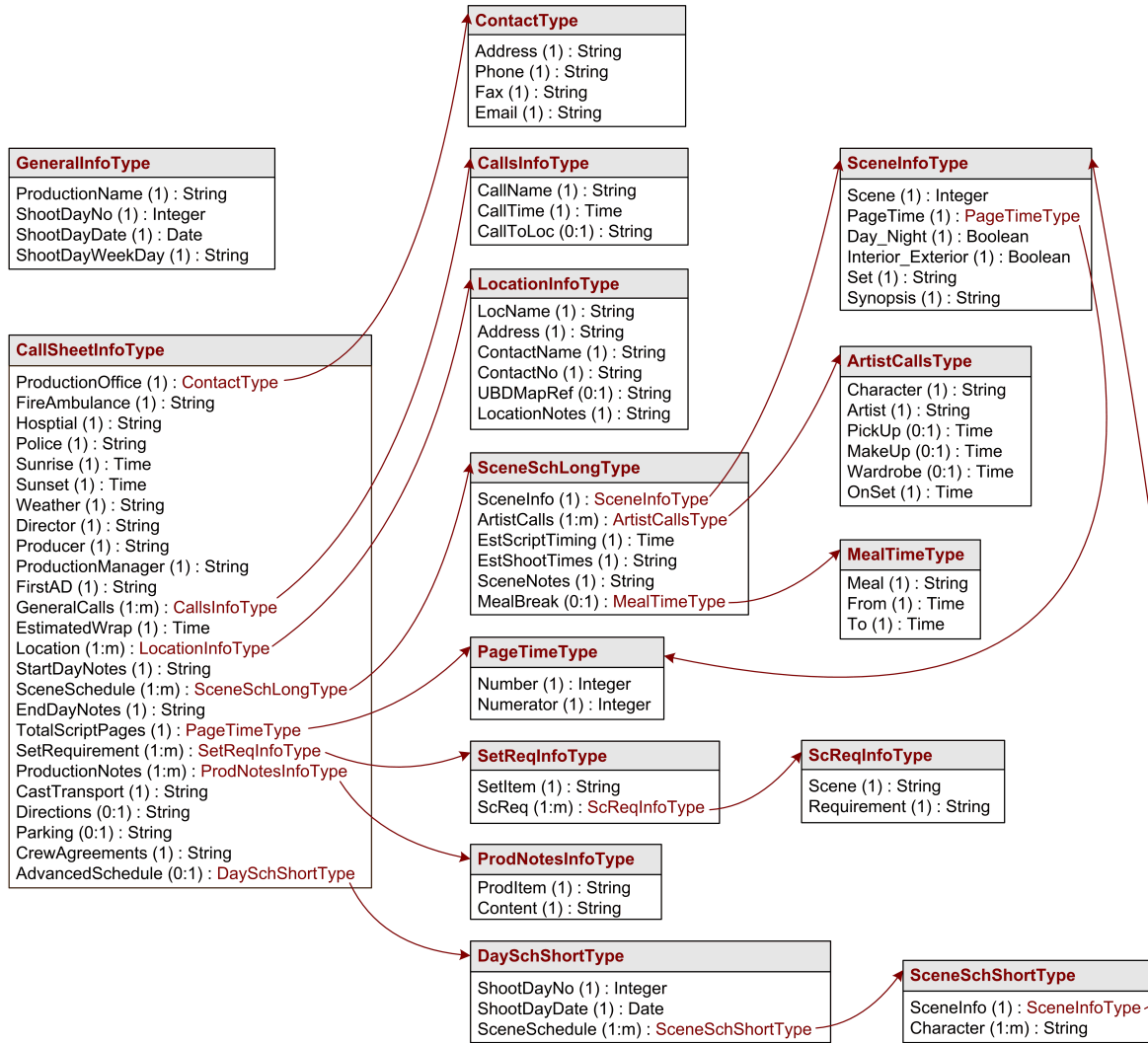


Figure 2: Schematic representation of the data types for specifying a call sheet.

The data manipulation mechanism in YAWL supports document processing and report generation in film production processes. Working data is stored in net variables that can be accessed and/or updated by all the tasks within the process. The data consumed and produced by a work item is captured by means of task variables associated with the corresponding task. Data transfer between net variables and task variables is specified by inbound and outbound mappings,

which are defined using the XQuery language. Fig. 3 depicts how task *Update Call Sheet* handles data at runtime. There are three task variables: *GeneralInfo* (input only), *CallSheetInfo* (input and output), and *Finalise* (output only). When a work item of type *Update Call Sheet* is checked-out, the values of its input variables are determined from the contents of the net variables (*General* and *CallSheet*) via input XQuery expressions. An example of the data extracted by this inbound mapping is shown inside the task symbol of *Update Call Sheet*. Later, when the work item is checked-in, the output variables of the task are used to update the corresponding net variables (*CallSheet* and *Finalise*) via output XQuery expressions.

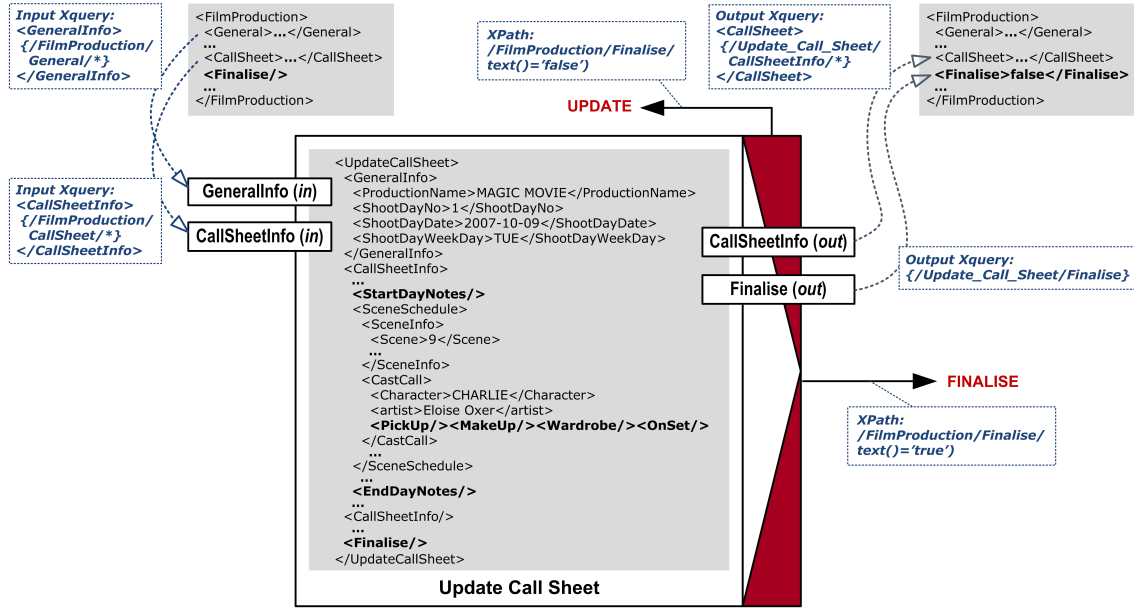


Figure 3: Data manipulation by task *Update Call Sheet*.

The data that the process engine supplies to the above work item of type *Update Call Sheet* is used to populate a Web form for the call sheet (see Sect. 3.4). Using this form, the user may perform updates to the call sheet, such as inserting “start-of-day notes” and he/she may indicate whether to finalise the call sheet (final submission) or to keep updating it (partial submission). This decision is captured in the output parameter *Finalise*. When the work item is checked-in later, the updated call sheet and the value of *Finalise* are stored in the net variables. The value of *Finalise* is then used to determine which outgoing flow of the XOR-split will be taken. The predicates associated such outgoing flows are specified by XPath expressions.

3.3 Resource Perspective

During a film production, each task is performed by one or more specific roles assigned to it. Examples are: a camera sheet is filled by the camera assistant, a sound sheet is filled by the sound recordist, a call sheet can be updated by a production coordinator or a production manager, etc. This allows to specify at design-time one or multiple roles to which instances of the task will be distributed at runtime, i.e. role-based task distribution to human resources. Table 2 lists the participants of a film production process with their user identifier, name and roles. Each participant has default password ‘yaw!’ (and the YAWL administrator has user identifier ‘admin’ and password ‘YAWL’).

User id	First name	Last name	Role
awhite	Alice	White	Production Manager
blaumen	Ben	Laumen	2nd AD
csmith	Cheryl	Smith	1st AD
cdouglus	Claire	Douglus	Editor
flang	Florian	Lang	Executive Producer
gwilson	Grace	Wilson	Production Assistant
hnewman	Harry	Newman	Sound Record
jlee	Jane	Lee	Production Coordinator
jgreen	Jay	Green	Continuity
jjason	John	Jason	Director
mwest	Michael	West	Producer
traymond	Thomas	Raymond	Camera Assistant
vrain	Vincent	Rain	D.O.P.

Table 2: Participants of a film production process – User ID, Name and Role.

3.4 Customised Forms

Most of the manual tasks in the film production process are performed to record and gather information for specific film production documents. They require input from the user by means of forms. YAWL supports automatic form generation for each work item, and the dynamically generated forms follow one standard (default) template. In order to support templates used in professional filmmaking, we chose to create custom Web forms and to link these forms to the worklist handler through forms connection interfaces. At design-time, this can be done by supplying the URL of a custom Web form to the decomposition of the corresponding task (‘Set Custom Form...’ option of a Task component in the YAWL Editor).

The custom forms were developed using standard Java technology. Item lists that appear in the forms are dynamically handled via Asynchronous Javascript and XML (AJAX), allowing users to insert or drop items in a lightweight manner. For example, Fig. 4 depicts the Web form ‘Crew List’ for task *Input Crew List*. On each form, a user can *load* an XML file (complying with the schema of the work item), *save* his/her input into a local XML file, *print* a copy of the form, and *submit* the form back to the worklist handler once it has been completed. Upon submission, a backup copy of the data on the form is also stored into the server (that installs the YAWL system).

Moreover, each form provides *data validation* upon save and submission to prevent the generation of invalid XML documents. This first stage of validation, realized via JavaScript on the client-side, is interactive: any field of the form which has been filled out with invalid data is reported to the user with suggestions for correction. This function is particularly useful when the forms are very complex and thus error-prone. The second stage of validation is provided by default by the engine on the server-side, and is not interactive. This is used to prevent the engine from processing invalid data that would block the execution of the process.

Finally, the ‘Call Sheet’ form provides an additional function – *print-preview* – which allows the user to generate a printer-ready document from the Web form, that resembles the hard copy format used in practice. The print-out version can then be distributed to the designated cast and crew in the film production team. The print-preview function relies on XSL transformations to convert the XML of the form to HTML.

Crew List

General

Production:

Crew Details

Role	First Name	Last Name	Contact No.	Email
Production Manager	Alice	White	0400123456	alice.white@magicmovie.com
1st AD	Cheryl	Smith	0401123456	cheryl.smith@magicmovie.com
2nd AD	Ben	Laumen	0402123456	ben.laumen@magicmovie.com
Camera Assistant	Thomas	Raymond	0403123456	thomas.raymond@magicmovie.com
Sound Recordist	Harry	Newman	0404123456	harry.newman@magicmovie.com
Continuity	Jay	Green	0405123456	jay.green@magicmovie.com
Director	John	Jason	0406123456	john.jason@magicmovie.com
D.O.P.	Vicent	Rain	0407123456	vicent.rain@magicmovie.com
Editor	Claire	Douglus	0408123456	claire.douglus@magicmovie.com
Producer	Michael	West	0409123456	michael.west@magicmovie.com
Prod Coordinator	Jane	Lee	0410123456	jane.lee@magicmovie.com
Prod Assistant	Grace	Wilson	0411123456	grace.wilson@magicmovie.com
Executive Producer	Florian	Lang	0412123456	florian.lang@magicmovie.com
Electrics	Steven	Hinggis	0413123456	steven.hinggis@magicmovie.com
Grips	Mathew	Landson	0414123456	mathew.landson@magicmovie.com
Standby Props	Martin	Glitter	0415123456	martin.glitter@magicmovie.com
Best Boy	Ryan	McCracken	0416123456	ryan.mccracken@magicmovie.com

Select a file to upload :

Figure 4: An example of a customised user form – crew list.

4 Setup

System Requirements: YAWL needs to be installed with Java 1.6 onwards to allow the custom forms of YAWL4Film to work (as they use JAXB technology).

YAWL4Film has been tested to be executed by any YAWL 2.1 engine release (YAWL4Study, YAWL4Enterprise and YAWLive). This section provides instructions on how to successfully carry out a film production process.

First we need to import the organisational data associated with a film production process into the yawl database:

1. Start the YAWL engine
2. Open the Control Centre and log in as *administrator* (id *admin*, password *YAWL*)
3. Click on *Org Data*
4. If the YAWL database has already been populated with resources, you can export them by clicking on the icon “Export Org Data to file” on the top-right corner

5. Click on the icon “Import Org Data from file” on the top-right corner and select the file *yawl4film.ybkp* from the directory
[YAWL Installation folder]/misc/examples/yawl4film/

A message box at the bottom informs us that the operation has succeeded by listing the number of resources that have been imported in terms of participants, groups, capabilities and positions.

Now we are ready to launch a case of a film production process:

1. Click on *Cases* in the Control Centre
2. Upload the file
filmproduction.yawl from the directory
[YAWL Installation folder]/misc/examples/yawl4film/
3. Select the loaded specification *YAWL4Film_Process*
4. Click on *Launch Case*

We have successfully launched a case of a film production process. Now we may log out as administrator and log in as a production manager (e.g. id *awhite*, password *yawl*) to view and edit the first work item *Welcome to Start Process*. Fig. 5 shows a screenshot of the corresponding user form. The user needs to provide the name of the movie for production, e.g. 'Magic Movie', and starts the production tasks.

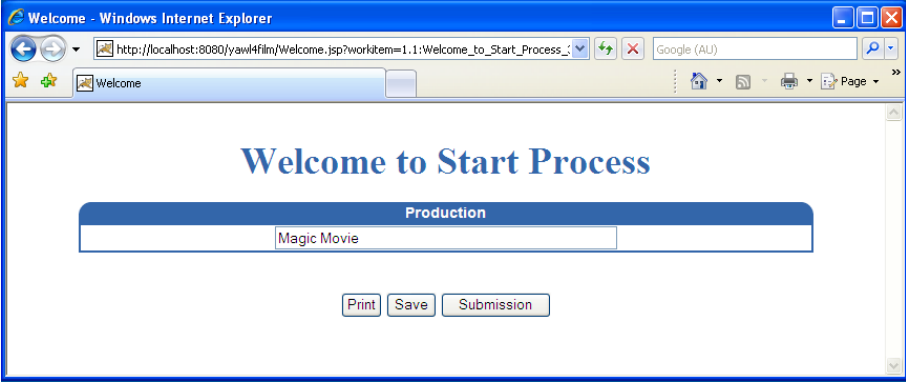
The image shows a screenshot of a web browser window titled 'Welcome - Windows Internet Explorer'. The address bar shows 'http://localhost:8080/yawl4film/Welcome.jsp?workitem=1.1:Welcome_to_Start_Process_'. The page content features a large heading 'Welcome to Start Process' in blue. Below this is a blue-bordered box with the title 'Production' in white. Inside this box is a text input field with the text 'Magic Movie'. Below the input field are three buttons: 'Print', 'Save', and 'Submission'.

Figure 5: A screenshot of ‘Welcome to Start Process’ form.

Since each task is linked to a customised user form which requires certain/large amount of data entry, we provide sample data (as .xml files) to facilitate users to run through a case of the film production process. The list of .xml files are available from the directory
[YAWL Installation folder]/misc/examples/yawl4film/sample_data/ and can be uploaded into the corresponding custom forms through *load* function. For example, file ‘CrewList.xml’ can be uploaded into form ‘Crew List’ (see Fig. 4).