A JDF-enabled Workflow Simulation Tool

Claes Buckwalter*

Keywords: Workflow, software framework, simulation, JDF

Abstract: Job Definition Format (JDF) is a relatively young industry standard that specifies a workflow model and a data format for describing print jobs and exchanging production information. Although most vendors today offer JDF-enabled products, little research has been conducted aiming at measuring the benefits of using JDF in production. One reason for this is likely the lack of suitable tools.

This paper presents a JDF-compliant tool that can be used to simulate the interaction of arbitrary devices in a print production workflow. This tool has many usage scenarios, one of them being the possibility to simulate devices in "live" production in order to measure JDF performance and detect production bottlenecks.

1 Introduction

A print production workflow can be modeled by a network of inter-connected distinct processes. A process is here defined as a sequence of related operations that consumes, transforms or combines input resources in order to generate output resources. For instance, the process of traditional offset printing takes input in the form of substrate, printing plate, ink, and press configuration parameters; and produces output in the form of printed matter. In a modern print production workflow a majority of the processes are represented by software systems that control and execute underlying subsystems, such as physical devices, for example a printing press. To achieve an optimized workflow, the software systems representing the workflow's processes need to communicate and exchange information in a well-defined manner. *Job Definition Format (JDF)* is a specification (CIP4, 2004a) that defines this information exchange between processes. The JDF Specification defines:

- A job ticket format for describing print jobs and all the processes and resources in prepress, press, and postpress required to produce print jobs

* Department of Science and Technology (ITN), Linköping University, Sweden

- A communication protocol, named *Job Messaging Format (JMF)*, for exchanging information between systems in the production workflow

To summarize, JDF specifies a solution for integrating all systems in the print production workflow, from prepress, through press, to postpress, including the management information systems (MIS) monitoring and controlling the workflow.

Under certain circumstances it may be of interest to be able to simulate one or more processes in a print production workflow. By inserting a simulated process into a real print production workflow one could test the effects that the added process has on the performance of the workflow and production as a whole. One could also build an entire print production workflow using only simulated processes, but use real management information systems to track and control these processes. Such a setup would allow one to experiment with different workflow configurations using real MIS systems.

The difficulties of designing a simulation tool for newspaper production workflows have been discussed in Nordqvist and Fällström (1996). A majority of the difficulties discussed also apply to general prepress, press and postpress workflows. However, by leveraging JDF a significant amount of these difficulties can be eliminated. The work of defining the process or processes to simulate and specifying how to integrate the simulation tool with real production systems is fully covered by JDF. Hence, once a software framework implementing the JDF Specification is in place, research efforts can be focused on developing sophisticated and realistic simulations of processes.

This paper presents a first version of a software framework that implements the JDF Specification and that can be used by a simulation tool to integrate with other JDF-enabled systems in print production workflows. The software framework presented provides an application programming interface (API) defining the services required of a JDF-enabled system. The framework, named *the Elk Framework*, can be used by developers wishing to implement a system that integrates with other systems using the JDF job ticket format and the JMF messaging protocol. As a proof-of-concept, a simulation tool that uses the Elk Framework is also presented.

2 Print Production Workflow

As described in Enlund (1998), a print production workflow, or any workflow for that matter, can be modeled as a network of linked activities that perform operations on objects. When an *activity* executes it consumes, transforms, or combines input *objects* in order to generate output objects. Applied to the activity of offset printing, input objects would be paper, ink, printing plates; output objects would be printed sheets of paper. An activity requires a set of *resources* in order to execute. Examples of resources required for printing are a printing press, and a worker operating the press. During execution an activity is under the influence of *control signals*. Input control signals modify or query the internal state of an activity. Output control signals are triggered by changes of the activity's internal state.

Activities are linked together using input objects and output objects to create process networks. An output object from one activity is the input object for the next activity in the workflow. Objects can link activities together in several different ways to produce networks of *parallel*, *sequential*, or *iterative* activities.

The systems that a print production workflow consists of can be grouped into two different types, as shown in Figure 2-1:

- Production systems on the plant floor that execute one or more activities in the workflow; for example, preflight software, a printing press, or a folding machine. Production systems are often controlled by software, and in some cases, often in prepress, consist entirely of software.
- Systems that monitor and control the production systems, for example a production planning system. These systems are computer systems and are usually called management information systems (MIS).

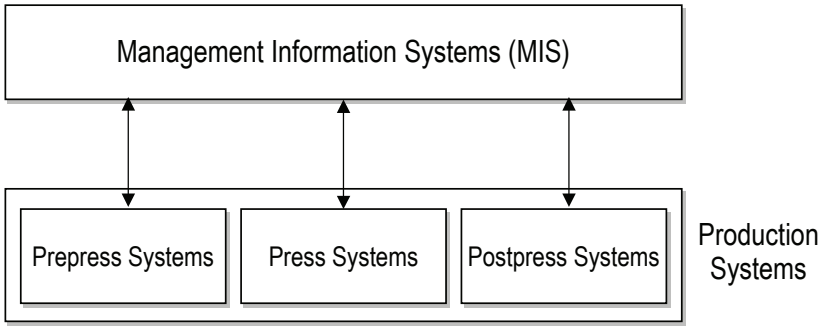

Figure 2-1 Systems in a print production workflow

A critical requirement for achieving an optimal print production workflow is that management information systems and production systems can communicate and exchange information in real-time during production (Enlund, 1998). Over the years, several industry standards have been developed that specify the integration of systems in the print production workflow; for example, *IFRAtrack* (Fällström, 2002), for tracking newspaper production, and *Print Production Format (PPF)* (Daun et al., 1998), for exchanging configuration data between systems. However, none of these standards have covered the entire workflow. It was not until the printing industry's major vendors joined forces and developed *Job Definition Format (JDF)* that there was an industry standard that covered all aspects of the print production workflow.

2.1 Job Definition Format (JDF)

Job Definition Format (CIP4, 2004a) is an industry standard maintained by an international consortium named *CIP4* (CIP4, 2005), consisting of system vendors, organizations, and research institutes. JDF defines a model for specifying a print job and all of the workflow activities required to produce a print job. The workflow model defined by JDF is similar to the generic workflow model described above. However, the terminology is slightly different (see Figure 2-2):

- The generic workflow model's a*ctivity* is called a *Process*.
- The generic workflow model's *objects* and *resources* are generalized and called *Resources*.
- The generic workflow model's *control signals* are realized using a messaging protocol called *Job Messaging Format (JMF)*. A system that executes a Process in a JDF workflow uses JMF to communicate with other production systems and management information systems in the workflow.

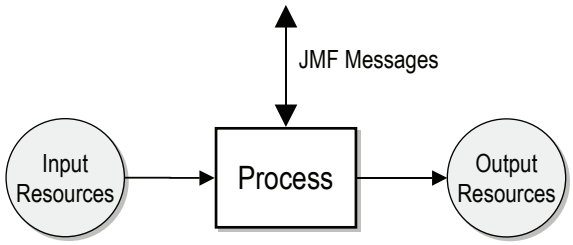

Figure 2-2 A JDF Process

In addition to defining a workflow model, JDF also attempts to define all of the Processes and Resources required in a print production workflow, such as *Screening* and *SaddleStiching* (Processes), and *ColorCorrectionParams* and *Employee* (Resources).

However, most importantly—JDF defines what information should be exchanged between systems in the workflow and the manner in which this information is exchanged, thus making it possible to connect and integrate all systems in a print production workflow.

2.1.1 JDF Integration Technologies

On a more concrete level, JDF's model of a print job, including the Processes required to produce the print job, is described using XML syntax (Yergeau et al., 2004). A JDF XML document that describes a print job is called a *JDF job ticket* or a *JDF instance*. A print job is produced in a JDF-enabled workflow by letting management information systems route the print job's JDF job ticket to the production systems that realize the Processes defined in the JDF job ticket.

To enable JDF job tickets to be transferred between the systems in a workflow JDF defines a messaging protocol called *Job Messaging Format (JMF)*. JMF messages are small XML documents that are used to control and monitor production systems, and to send and track JDF job tickets through the workflow.

The strength of JDF, integrating the heterogeneous systems in the print production workflow, can naturally also be applied to integrating simulation tools into real production environments.

3 Process Simulation

Under certain circumstances it may be of interest to be able to simulate one or more processes in a print production workflow. By inserting a simulated process into a real print production workflow one could, for example, test the impact the added process has on the workflow. One could also build an entire print production workflow using only simulated processes, but use real management information systems to track and control the processes in the workflow. Such a setup would allow one to experiment with configurations of production systems using real management information systems.

In order to simulate a process one must first create a model of the process. The type of simulation model most suitable for modeling a process in a print production workflow is a discrete event simulation model (Bäck et al., 1997). Once a process simulation model has been implemented in software, the model must be integrated with a print production workflow. JDF specifies how to integrate systems in a print production workflow. Hence, integrating a simulated process with real systems in a print production workflow is trivial if the simulated process has an integration layer that implements the JDF Specification.

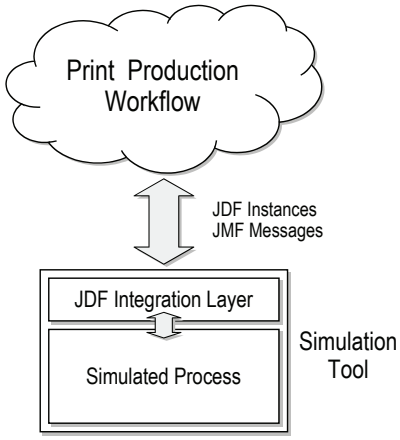

Figure 3-1 Process Simulation with JDF Integration

A generic and reusable software framework that provides the services required of the JDF Specification would be extremely useful for integrating simulated processes with real production systems. The remaining part of this paper describes such a framework.

4 The Elk Framework

The JDF Specification defines several roles that a system in a workflow can have (CIP4, 2004a, 2004b). Two of these roles are the Device role and the Agent role. A system with the *Device* role reads a JDF job ticket and executes one or more of the processes specified in the job ticket. A system with the *Agent* role can create and modify JDF job tickets. If a system has both of these roles it implements a *Worker Interface* (CIP4, 2004b). A system that implements the Worker Interface is called a *Worker*. More specifically, a Worker is a system that can receive a JDF job ticket; execute one or more of the processes specified by the job ticket; update the job ticket; and send the job ticket to another system in the workflow, see Figure 4-1. A JDF-enabled process simulation tool would need to implement a Worker Interface, and would therefore be Worker.

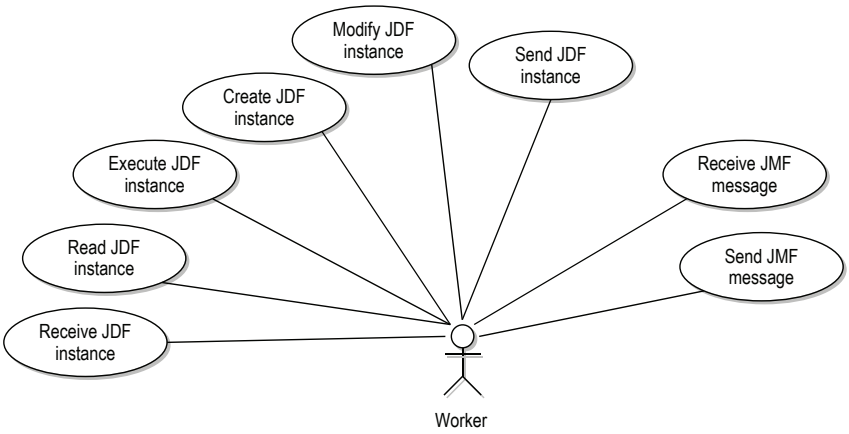

Figure 4-1 Worker use case diagram

The generic functionality required of a Worker can, as Figure 3-1 hints, be abstracted into a reusable software framework for JDF integration. Presented here is such a framework, named t*he Elk Framework*.

The Elk Framework is implemented in the Java programming language and consists of two parts:

- An application programming interface (API) that expresses the generic functionality required of a Worker as a set of services
- A reference implementation of the API

4.1 Services

The Elk Framework API consists mainly of interfaces and abstract classes. In other words, the API does not contain actual implementations of any services, it just defines the service and how they are accessed using the Java programming language. One can say that the API is a specification of the services. The following are the most important services defined by the Elk Framework API:

4.1.1 Messaging Gateway

A *Messaging Gateway* is used to send and receive JMF messages and JDF instances without exposing the underlying network transport protocol. By using a Messaging Gateway one needs not be concerned with the details of sending and receiving JMF messages and JDF instances. Instead one can focus on processing the JMF/JDF received and generating JMF/JDF to be sent, see *Message Processor*.

4.1.2 Message Processor

A Messaging Gateway routes JMF messages to Message Processors. A *Message Processor* knows how to process a specific type of JMF message, for example a *QueueStatus* JMF message (CIP4, 2004a), which queries the status of a Worker's queue. A Message Processor processes JMF messages in some implementation specific way, possibly delegating the processing to other components within the Worker.

4.1.3 Subscription Manager

Management information systems monitor a Worker by subscribing to notifications of state changes that occur within a Worker. A *Subscription Manager* alleviates the Worker from having to deal with the intricacies of managing subscriptions and sending JMF messages containing notifications of state changes.

4.1.4 Job Queue

A *Job Queue* manages jobs, corresponding to JDF instances, sent to a Worker for processing. Jobs are typically added to a Job Queue via a *Message Processor*. A *Process*, see below, takes jobs from a Job Queue and executes the jobs.

4.1.5 Process

A *Process* component reads JDF instances and executes one or more of the processes defined in a JDF instance. In other words, a Process is the component in a Worker that communicates directly with the underlying software or equipment that does the actual work. For example, a Process component that implements the *ConventionalPrinting* process (CIP4, 2004a) would configure the printing press according to the parameters specified in the JDF instance and then start the printing press.

4.2 Reference Implementation and Simulation Tool

Besides an API, the Elk Framework also consists of a reference implementation that exemplifies how the API's services can be combined to create the JDF integration layer of a Worker. By leveraging the reference implementation a simulation tool can be integrated with a JDF-enabled print production workflow, see Figure 3-1. All that is required is that one creates an implementation of a *Process* component, see 4.1.5, that executes the process simulation model.

As a proof-of-concept a simulation tool has been created that uses the Elk Framework. The tool implements a simple simulation of a proofing process, in JDF terminology called *Approval* (CIP4, 2004a). The Approval process is used when a resource, for example a print-ready PDF

document, needs to be proofed/approved before the next process in a print job's workflow can commence.

Figure 4-2 shows an overview of the most important components that the simulation tool consists of. The component *Simulated Approval Process* implements the simulation model. It takes a job, a JDF instance, from the job queue; simulates the execution of the job, consuming input resources and producing output resources accordingly; and then sends off the updated JDF instance for further processing in the workflow. During process execution, all state changes within the simulation model are sent to the management information systems monitoring the process or tracking the job.

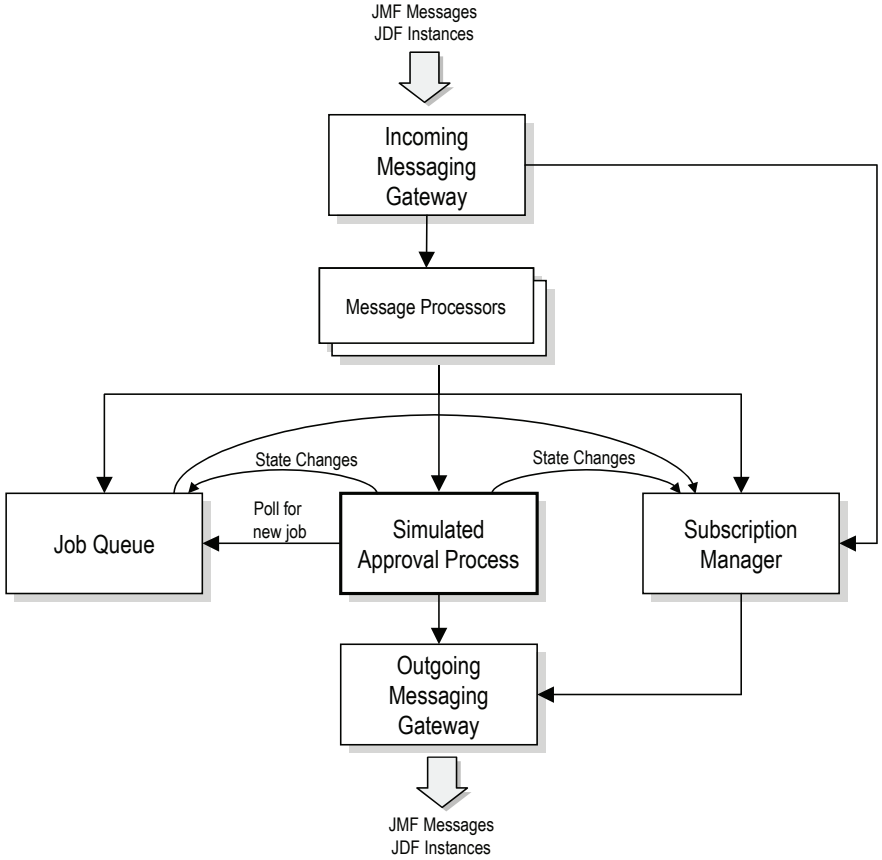

Figure 4-2 Overview of the simulation tool's components

5 Conclusions and Future Work

Job Definition Format (JDF) defines a workflow model for print production and specifies how to integrate the heterogeneous systems in the print production workflow. By leveraging JDF, the integration of simulation tools into real print production workflows becomes greatly simplified. A simulation tool into two distinct modules: a module concerned with simulating the production process and a module that provides the services necessary for JDF integration. This paper focused on the JDF integration module and presented the Elk Framework—a software framework that provides the generic services required by any system that wishes to integrate with a JDF-enabled print production workflow.

As a proof-of-concept, a simple simulation tool was presented that uses the foundation and infrastructure provided by the Elk Framework. Preliminary testing of the simulation tool in an environment with commercial production systems has been conducted in cooperation with the CIP4 Organization. The preliminary tests have been successful and further tests will be conducted as the Elk Framework evolves.

The development of the Elk Framework will be continued with a future focus on designing and implementing more accurate simulation models of production processes and running these simulation models in real print production environments. Of interest is also the possibility of creating a virtual print production workflow consisting of simulated processes.

More information regarding the Elk Framework can be found at the project web site: *http://elk.itn.liu.se*

6 Acknowledgments

The author wishes to thank professor Björn Kruse, ITN, Linköping University, and Dr. Rainer Prosi, CIP4, for their valuable feedback and support. The author would also like to extend his gratitude to the CIP4 Organization, and the T2F research program (www.t2f.nu) for their financial support.

7 References

Bäck, A., Lehtonen, T., Karttunen, A., Kuusisto, M., Launonen, R.
1997 *Evaluation of Printing Production in Networks by Means of Computer Simulation Models*, Advances in Printing Science and Technology, vol. 24, pp. 177–190

CIP4
2004a *JDF Specification Release 1.2*, http://www.cip4.org/documents/jdf_specifications/JDF1.2.pdf

2004b *Base Interoperability Conformance Specification (Base ICS) 1.0*, http://www.cip4.org/document_archive/documents/ICS-Base-1.0.pdf

2005 *The International Cooperation for the Integration of Processes in Prepress, Press and Postpress Organization (CIP4)*, http://www.cip4.org

Daun, S., Lucas, G., Schönhut, J.
1998 *CIP3 PPF Specification Version 3.0*, Darmstadt, Germany, http://www.cip4.org/documents/technical_info/cip3v3_0.pdf

Enlund, N.
1998 *Workflow Management in Graphic Arts Production*, Advances in Printing Science and Technology, vol. 25, pp. 191–203

Fällström, F.
2002 *IFRAtrack 3.0*, IFRA Special Report 6.21.3, Darmstadt, Germany

Nordqvist, S., Fällström, F.
1996 *Simulation of Newspaper Production Processes—Decision and Management Support Tools*, TAGA 1996 Conf. Proc., pp. 299–317

Yergeau, F., Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler E.
2004 *Extensible Markup Language (XML) 1.0 (Third Edition)*, http://www.w3.org/TR/2004/REC-xml-2004020