Alces – A Tool for Testing JDF Compliance

Claes Buckwalter
Media Technology, ITN, Linköping University
Norrköping, Sweden

Abstract

Job Definition Format (JDF) is an industry standard that specifies a data format for describing print jobs and exchanging production information between systems in the print production workflow. Implementing JDF support in a device, such as a digital printing press, is a complex task. Developers need tools for testing that their implementations conform with the specifications provided by CIP4, the industry consortium that maintains the JDF standard. This presentation will introduce *Alces* – a tool for testing the JDF conformance of devices. Alces connects and sends JMF messages to a device and executes test cases on all data returned by the device. Alces can be run in automated mode suitable for regression testing, or in interactive mode suitable for manual user-controlled testing.

Introduction

Although most vendors in the printing industry offer some level of JDF-support in their products there is unfortunately no guarantee that JDF-enabled products from one vendor will be interoperable with those from another vendor. To alleviate this problem the CIP4 organization has developed the *Interoperability Conformance Specification (ICS)* documents. Each ICS document defines what parts of the JDF specification [1] that a certain class of product is required to implement. The Base ICS [2] is the least common denominator that all JDF-enabled products must comply with. On top of the Base ICS there are at the time of writing six other ICS documents [3] for different product classes, such as MIS, prepress, and integrated digital printing.

Testing JDF Compliance

Testing and verifying the behavior of a JDF compliant product is a complex task. For example, testing that a printing press interprets a JDF job ticket and produces the correct output would involve analyzing the printed sheets and in some way attempting to verify that they match what the JDF job ticket specified. There are many factors that have nothing to do with a product's ability to interpret JDF data that can influence the physical output produced by the product. Functional testing of this kind is beyond the scope of the test tool presented in this paper. Instead, the test tool presented here is concerned only with testing a product's ability to consume and produce JDF instances and JMF messages as specified by the ICS's that the product claims to comply with.

Alces

The test tool, *Alces*, implements the role of a *manager*; a piece of software that sends JDF instances and JMF messages to a *worker*. The worker role is implemented by a piece of software that processes received JDF instances and JMF messages and after processing may send information back to the manager. An example of a product with the manager role is a prepress controller that sends jobs to a RIP, the RIP being a worker. In our test scenario Alces is a manager and the software being tested by Alces is a worker.

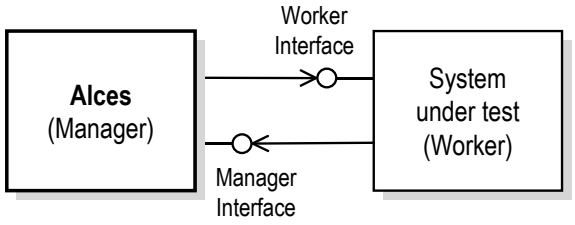

Figure 1. Manager-Worker

Test Procedure

Alces sends JDF/JMF data to a worker and tests that the worker responds correctly with valid output. The following tests are performed on JDF instances and JMF messages produced by the worker:

1. The syntax of the XML is validated using CIP4's *CheckJDF* [5]
2. The presence and correctness of all required XML elements and attributes specified by the relevant ICS documents is checked
3. The XML data is examined and checked that it is correct according to the current context

Validation using CIP4's CheckJDF tool is straightforward and includes validation against the JDF XML Schema [4]. The presence of the XML elements and attributes required by the ICS documents are tested using JDF *device capability* definitions [1]. One test case with a set of device capabilities files is available for each ICS. Verifying that JDF instances and JMF messages produced by a worker are correct according to the current context is performed by maintaining a record of all JDF and JMF data previously consumed and produced by the worker. Any new data produced by the worker is compared with the previously recorded data and validated using specifically built test cases.

All test results are logged in an XML file. It is left to other tools beyond the scope of this paper to analyze and visualize the information in the XML log file. For

example, the XML log file could be transformed using XSLT to an XHTML document for viewing in a web browser.

Architecture

Alces is implemented as a Java application that consists of three main modules, as shown in Figure 1. The client module sends JDF/JMF data to the worker under test; the server module receives JDF/JMF data from the worker; and the test module coordinates the test data sent to the worker and validates all data produced by the worker. The outgoing test data used to probe the worker consists of a set of manually written JDF and JMF files; all validated using CIP4's CheckJDF tool [5]. These files can easily be modified or replaced if necessary. The test cases are dynamically loaded at startup of Alces and are configured using configuration files. New ICS support can be added by creating a new test case.

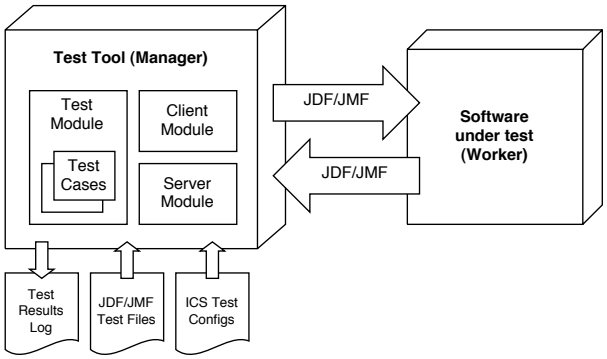

Figure 2. Architecture

Future Work

Alces has so far been used for testing CIP4's internal software tools but more extensive testing against third party products must be conducted in order to thoroughly evaluate Alces. Currently, there is only a test case for the Base ICS. Additional test cases for other ICS's must be implemented for testing more than the minimal level of JDF functionality. Nonetheless, Alces is still useful since vendors will need to test their Base ICS implementations.

The complete source code and up to date information regarding the development of Alces is available at [6].

Acknowledgements

The author thanks Ola Stering and the members of CIP4 for testing Alces and giving valuable feedback. A word of gratitude also goes to the author's supervisor Professor Björn Kruse, ITN, Linköping University, for his support and encouragement.

References

1. JDF Specification Version 1.2 [Online]. Available: http://www.cip4.org/documents/jdf_specifications/JDF1.2.pdf [2005, September 13]
2. Base Interoperability Conformance Specification (Base ICS) [Online]. Available: http://www.cip4.org/document_archive/documents/ICS-Base-1.0.pdf [2005, September 13]
3. ICS Registry [Online]. Available: http://www.cip4.org/document_archive/ics.php [2005, September 13]
4. XML Schema for JDF Version 1.2 [Online]. Available: http://www.cip4.org/Schema/JDFSchema_1_2/JDF.xsd [2005, September 13]
5. JDF Open Source [Online]. Available: http://www.cip4.org/open_source/index.html [2005, September 13]
6. Alces [Online]. Available: http://elk.itn.liu.se/alces [2005, September 13]

Biography

Claes Buckwalter has a Master's degree in Media Technology and Engineering from Linköping University, Sweden, where he is currently a Ph.D. student. Claes's Ph.D. studies focus on integration technologies and process automation in the graphic arts industry. Claes is also involved in the CIP4 organization where he is chairman of the Tools & Infrastructure working group.