KNOWLEDGE REPRESENTATION FOR DIGITAL PUBLISHING WORKFLOW

By

Amado E. Pereira-Rangel

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS

May, 2006

Approved by:

Wilson Rivera, Ph.D Member, Graduate Committee	Date
Nayda Santiago , Ph.D Member, Graduate Committee	Date
Fernando Vega. , Ph.D President, Graduate Committee	Date
Jose Cruz, Ph.D Representative of Graduate Studies	Date
Isidoro Couvertier, Ph.D Chairperson of the Department	Date

Abstract of Dissertation Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Master of Science

KNOWLEDGE REPRESENTATION FOR DIGITAL PUBLISHING WORKFLOW

By

Amado E. Pereira-Rangel

May 2006

Chair: Fernando Vega
Major Department: Electrica and Computer Engineering

Digital Publishing involves printing processes where the film and plate making stages are eliminated. Digital printing is a process prone to error and subject to trial and experimentation; therefore resources such as ink, paper, and time are wasted before a top-quality output job is obtained. To fully exploit the potential of Digital Publishing flexible workflows are needed to provide services comparable to traditional publishing, but with the value added by the digital technology. Decision-making in such complex settings is highly dependent on human expertise. Knowledge engineering can play an important role in representing critical knowledge and performing inferences for decision-making that can lead to the development of knowledge-based system for automated workflow management. A framework for decision-making in workflow management is developed. This framework consists of an ontology combined with a ruled based system. The ontology contains a model for digital publishing while the ruled based provides mechanisms for inferences on this model. A set of rules for a digital publishing workflow for a print shop with three types of printers was tested with three critical printing job scenarios. Tests showed

the validity of the approach with accurate problem diagnosis, recommendation for proper solution and explanations in all the scenarios. The framework is flexible and can be customized for other print shops settings and other job scenarios.

Resumen de Disertación Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de Maestría en Ciencias

REPRESENTACION DE CONOCIMIENTO PARA PUBLICACION DIGITALES EN FLUJO DE TRABAJOS

Por

Amado E. Pereira-Rangel

Mayo 2006

Consejero: Fernando Vega
Departamento: Ingeniería Eléctrica y Computadoras

La impresión Digital es un proceso propenso al intento, error y la experimentacin razon por la cual se pierden muchos recursos tales como tinta,papel y tiempo antes de lograr obtener trabajos en dispositivos de salidas como impresoras de alta calidad. La ingeniera de conocimiento puede desempear un papel importante en la representacin de conocimiento crtico y la ejecucin de las inferencias para la toma de decisin en la gerencia del flujo de trabajo en publicaciones digitales. En adicion, un sistema basado en el conocimiento puede proporcionar explicaciones y justificaciones de las decisiones sobre la gerencia del flujo de trabajo en en el proceso de prepresena. El desarrollo de un sistema basado en el conocimiento para el proceso del preprensa har posible solucionar los problemas de complejidad realista que requieren una cantidad significativa de conocimiento humano mediante el uso de diversas tcnicas de la representacin de conocimiento y toma de decisiones. Este sistema es una parte importante para lograr la autimatizacion en las publicaciones digitales.

Deditacted to:

My Family

My Mother-Mariana

My Fiancee Priscila

ACKNOWLEDGMENTS

I would like to thank a lot of people who helped me make this possible. First my commitee members, Thanks to Wilson Rivera for all his support and encouragement that kept me working in this Research; thanks for his support at PDC LAB, and for giving me the oportunity to work in this environment. Thanks to Fernando Vega for all his help and valuable contribution of this work. Thanks for giving me the opportunity to expand my knowledge in specialized area.

Thanks to HP in The Digital Publishing Program and HP Digital Publishing Philanthropy.

Thanks to my lab partners Wilson, Gustavo, John, Elliot, Rene, Juan. Thanks to my family; my sister, my mother and father for believing and supporting in me to accomplish my goals. Thanks to my friends Diego and Roman. Thanks to my Fiancee for helping me do this research, for her support and understanding.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

Figure page

LIST OF ABBREVIATIONS

DP	Digital Publising
RDF	Resource Description Framework
OWL	Web Ontology Language
XML	Extensive Markup Language
PDF	Portable Document Format
PS	PostScript
JDF	Job Definition Format
XMP	Extensible Metadata Platform
OIL	Ontology Inference Layer or Ontology Interchange Language
DAML	DARPA agent markup language
W3C	World Wide Web Consortium
JESS	a rule engine for the java platform
RDFS	Resource Description Framework Schema
CLIPS	c Language Integrated production system
RIP	ripping
ICC	international color consortium
RGB	is an additive model which consits of Red, Green and Blue
CMYK	a subtractive color model used in color printing Cyan, Magenta, Yellow and Black
PPD	PostScript Printer Description

CHAPTER 1
INTRODUCTION

1.1 Justification

Digital printing is a process prone to trial and error, and experimentation, therefore resources such as ink and paper that are utilized before a top-quality output may be obtained [1]. A digital publishing job has to go through a number of steps prior to being actually printed. These steps aim at ensuring the best quality for the job [2]. This process generates information that needs to be organized and structured to be able to make its best use of decision support for production flow, quality control and tracking. In digital publishing it is needed to restructure the metadata so it can have a certain level of formalism and structural value in the data. This is used for production flow purposes, and tracking and analyzing systems to generate a decisional support system based on knowledge.

A workflow management system is a framework that can create and manage the execution of processes through the use of software, which allows the interpretation of the process definitions. The workflow management system process is a representation of the steps that take place at a certain moment of a series of steps to achieve a certain goal. The workflow systems used by the industry are increasingly complex. The results of new applications for these workflow systems, such as electronic commerce, the need for "mass customizations" in "customer care", and the need for some parts of the workflow to operate immediately, is becoming imperative for the

knowledge employee that works on a certain process already defined in the systems.

In the digital publishing industry there are critical production steps where the knowledge of certain employees may be an essential part of the system, especially in the prepress stage. For this reason, it is important to provide and identify the key knowledge that certain employees need to use in order to help improve the decision-making process. Making information available to the workflow systems is adequate, but just a data perspective system is not enough for the decision making that can affect significantly the final outcome of a print job.

The development of a knowledge-based system for the prepress process in digital publishing makes it possible to solve some problems of realistic complexity that require a significant amount of human expertise.It is achieved through the use of different knowledge representation and decision-making techniques.

It is necessary to capture,store, access and manipulate the information that is generated in the processes that compose the Digital Publishing workflow. This framework must be developed creating an expert system and an ontology, which provides ways to store, query, and perform semantic reasoning with this information to make better decisions.

The cost of prepress is not known until the files have been examined. Generally prepress costs are fixed in the estimate and do not take into account actual problems that may occur with the files. If problems arise in the prepress stages, a press can easily sit idle waiting for the job. This downtime is unacceptable. Digital files should be examined before cost and time estimates are set.

Customer service agents must keep customers informed about the status of their jobs. Frequently, digital files sit in prepress for days before being examined. That is not acceptable either immediately or within hours of receiving a job. A customer must be notified of any problems.

While the process can be partially automated, there is currently no way to eliminate the preflight process. Printing companies who neglect to set time aside for this process will inevitably lose money on jobs if they choose to fix these problems without charging customers.

A careful approach must be taken when notifying customers of problems. Customers need to be educated as to what is wrong with files and what can be done to avoid problems in the future.

1.2 Problem Statement

The objective of this thesis is to develop an information system that helps to make decision support system, extracting and using relevant information from raw data, documents and personal knowledge less troublesome for the expert. This decision support software helps in problem solving and decision capabilities making to automate the workflow system, of the digital publishing environment.

1.3 Objectives of this Thesis

This research is focused in understanding the use of knowledge based systems, and semantic web application for decision-making support in workflow management systems for digital Publishing. To achieve this goal the following objectives were set:

- To define partially the metadata needed for digital publishing workflow

- To carryout a comparative study of knowledge representation and inference models for decision making in digital publishing workflow

- To develop a prototype framework for data analysis for decision support in digital publishing workflow

1.3.1 Expected Outcomes/Deliverables

The expected outcomes of this investigation are the following:

- Prepress process ontology

- A comparative study of knowledge representation and inference models for decision-making in digital publishing workflow

- A set of tools for metadata analysis for digital publishing workflow management, for search and storage of, and inference on relevant information for better decision-making in the digital publishing workflow

1.3.2 Contributions

The main contributions of this thesis are:

- An ontology for Digital Publishing

- Digital Publishing Knowledge sharing through the use of an ontology

- A framework for a Recommender System Development for Digital Publishing through the use of the ontology.

For experimentation and validation purposes a preflight process is going to be used as a case study. In this process, decisions should provide enough information for the workflow engine to route documents, in this case it is whether the PDF Job file can be printed or not. To assess the effectiveness of the system, a sample of critical jobs is going to be sent to the system to identify the proper solution in each case.

Next chapters exhibit literature review, which state all the technology and concepts used for the development of the knowledge base system for Digital Publishing.

CHAPTER 2
LITERATURE REVIEW

This chapter is an analysis of some relevant work that was used as a base for this thesis. The areas been analyzed are: a) Ontology, b) Knowledge Representation, c) Expert System, d) Digital Publishing

2.1 Ontology

Many definitions of ontology have been developed; even for AI literature ontology has different meaning. Its definition may vary depending on the field of study, its meaning is molded to satisfy the intended purpose of each area of study.

- Ontology defines the basic terms and relations comprising the vocabulary of a domain as well as the domain area,along with the rules for combining terms and relations to define extensions to the vocabulary[3]. But in a more general definition, ontology is an explicit specification of conceptualization.[4]. Ontology is also defined as a formal specification of a shared conceptualization[5]. It is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base [6].

  Many definitions have been used for ontology, but essentially all of them are domain theories that specify a specific domain of vocabulary entities, classes, properties, predicates and functions, and a set of relationships that hold among those vocabulary terms. Ontology provides a vocabulary to represent knowledge about a domain for describing specific situations[7].

Recent work on ontology development has mainly focused on building domain ontologies in inter-organizational contexts for of building the Semantic web. These domain ontologies can significantly improve knowledge management practices within organizations. Therefore, limited evidence and information are available on the opportunities and challenges that organizations face in building ontologies. Typical problems related to knowledge retrieval include the overwhelming amount of information, the dynamic nature of the information integration, and disparate and fragmented data sources.[8]

Ontologies are used to capture knowledge about some domain of interest. An ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different advantages. The most recent development in standard ontology languages are OWL and RDF from the World Wide Web Consortium W3C. Like Protege, OWL and RDF make it possible to describe concepts, it has a richer set of operators and/or negation. It is based on a different logical model, which makes concepts to be defined as well as described.

The concepts could even be referenced by different terms as long as they have the same semantics and mapping is provided. An ontology offers a uniform set of terms to refer to concepts throughout a domain. They are organized in line with how users conceive a domain rather than the data structures created to conveniently store the information. The ontology is used as a neutral interchange format as described by Uschold [9, 10] Complex concepts like description logic can therefore be built up in definitions out of simpler concepts. Furthermore, the description logic model allows the use of a reasoner, which can check whether or not all of the statements and definitions in the ontology are mutually consistent and which

concepts fit under which definitions. The reasoner can therefore help maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent.

Thus an ontology-based application heavily depends in the ontology to model the given domain. In the case of this thesis the domain is Digital Publishing Workflow. Due to the constant changes in the workflow systems or the users requirements, ontology will evolve over time, since these ontologies are complex, interwoven structures, and can be used to resolve many knowledge-based systems[11].

Existing methodologies and practical ontology development experiences have in common that they start from the identification of the purpose of the ontology and the need for domain acquisition. They differ in the steps that they follow, and the use of the ontology itself. In this thesis we integrate existing methodologies and lessons learned from practical and theoretical ways to develop an ontology and use it in a current application which will be explained in the next chapter.

Further in the thesis, the concept of ontology development process will be explained to achieve a knowledge base system capable to make decisions in a certain process.

2.2 Expert System and Decisional Support System

Rule-Based Expert Systems are advanced computer programs, which try to emulate the human reasoning with capabilities to solve problems , using knowledge within a particular discipline [12]. One of the most practical applications of artificial intelligence in business is the development of expert systems.[13]

Designing a successful expert system is a very challenging and demanding process. It has been stated that one of the central concepts, which lead to an improved expert system development, is the design process. It is very important that the expert system builder considers the concept of the design process at the outset of developing such systems.

Rule-based systems possess a certain heuristics that form the static knowledge base, as well as some inference and search processes. Some of the milestones found in the rule based system is that they are very complex and are related to specific domains. They would usually need a human expert that possesses the knowledge to be solve it.

The most important components of these expert systems are:

- Knowledge Base: long term memory suggests solutions to problems based on feedback provided by the user, and are capable of learning from experience.

- Database: working memory or short term memory for the expert systems.

- Inference Engine: derive answers from a knowledge base. It is the brain of the expert system that provides a methodology for reasoning about the information in the knowledge base and for formulating conclusions.

- User Interface: it affects the amount of effort the user must expend to provide input for the system and to interpret the output of the system.

- Auto Explanation Module: The response of the expert system to the question WHY is an exposure of the underlying knowledge structure.

– Strategy Module - conflict resolution.

– Knowledge Engineer - it is concerned with the representation chosen for the expert's knowledge declarations and with the inference engine used to process that knowledge.

These knowledge based systems try to obtain the proper solution to the adequate problem established by the Digital Publishing Workflow domain. It performs reason over the representation of the human knowledge.

2.3 Digital Publishing

Digital publishing [14] permits the linking of printing presses to computers, by eliminating the use of films or plates. As a result digital publishing has the potential to raise the quality level for short-run printing, and the printing of documents that are highly variable in data content and layout for each copy coming out off the press. However, the realization of this potential has, to date, been seriously hampered by a number of difficulties. These include both the problem of getting the document to print correctly without artifacts on the press and the difficulty of managing the increasingly complex workflow that results from shorter run jobs that must be completed in less time, and which require access to back-end databases for variable data content. Consequently, digital publishing not only opens up new business opportunities but also requires new business models, which lead to new workflow designs. The fact that information remains digital from the design stage all the way to printing leads to potential automation of processes that in traditional workshops are still manually executed. [2]

In the digital printing arena exist a number of commercial tools to enable the digital printing processes. These tools range from digital content management (e.g. Documentum and InterWoven), content creation (e.g. Adobe Photoshop, Adobe

Illustrator and Macromedia Freehand), and page layout design (e.g. Quark XPress, Adobe InDesign, Adobe PageMaker, and Adobe FrameMaker) to preflight (e.g. Enfocus Pitstop, Marzware FlightCheck, Extensis Preflight Pro, and Agfa's Apogee) and page proofing (Agfa's Sherpa and HPColorBlind). These tools provide certain level of automation on isolated and very specific digital printing processes therefore generating islands of automation. JDF is the print communication industry's most highly anticipated standard since Adobe(r) PostScript(r) and Portable Document Format (PDF). Design, prepress, and print professionals are looking at JDF[15] as an enabling technology to speed production, increase reliability, and enhance the quality and flexibility of the printed job. In an industry where cost and time efficiency are foremost, JDF promises to accompany in a new era of workflow automation. Using a subset of the data that JDF provides, a new system can be developed to manage this metadata and make it more useful for Digital Publishing Autoimmunization.

There also exist a number of commercial digital printing workflow engines (e.g. FreeFlow Digital Workflow Collection from Xerox and HP Production Manager), commercial general-purpose workflow engines (e.g. BEA's Web Logic Process Integrator, Blue Titan, and Biztalk), as well as open-source general purpose workflow engines (e.g. Xflow, OpenWFE, jBpm, and YAWL). These workflow engines lack an integrated view of pre-press stages in digital publishing workflows. The Print on Demand initiative (PODi) has released its newest Best Practices report [16] presenting new digital print case studies as well as best practice principles for creating highly effective digital print solutions. The report supply print service providers, vendors, marketers, editors, analysts and others of the most innovative ideas and proven strategies from today's digital marketplace. Nevertheless, the report also

illustrates the current need for new concepts and techniques to integrate and automated digital publishing workflows. This report will provide enough information for the expert system to make decisions whether that document can be printed or not, based on the development of a knowledge based system.

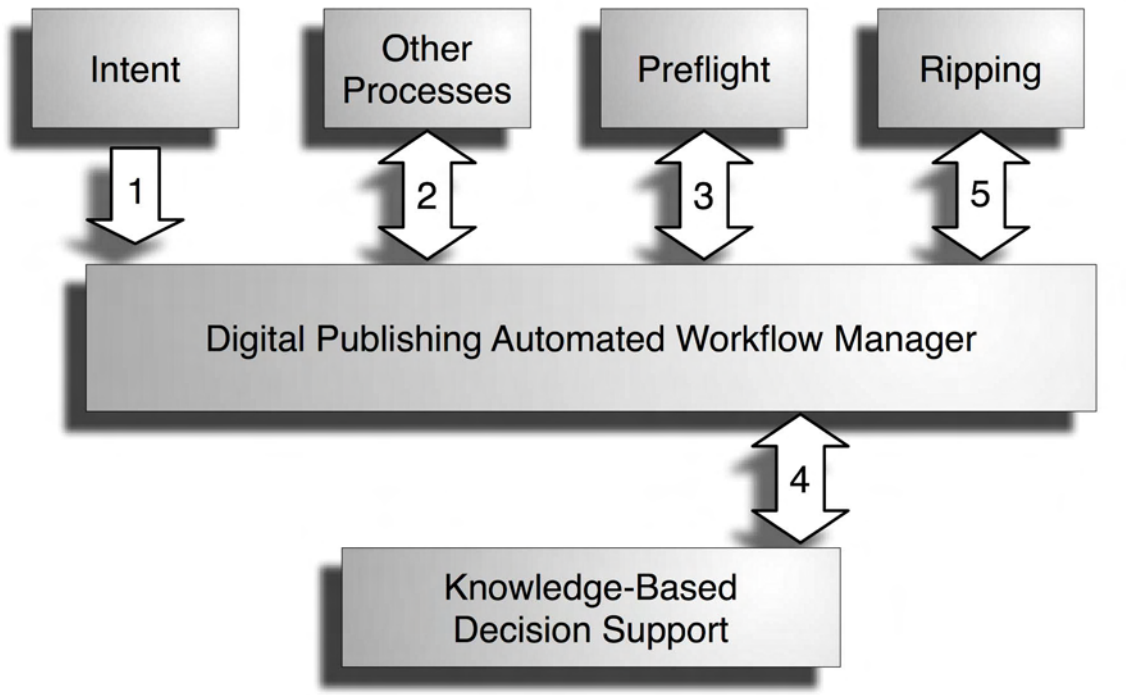

Figure 2–1: Digital Publishing Automated Workflow

As Shown in 2–1. in publishing there are many steps a job has to get through prior to be published. These steps ensure that the quality of the job is the best one for the type of job that will be published. One of those steps is preflight. Preflight helps to ensure that all required files (application, PostScript, or PDF files, fonts, and graphics) are included in the package sent to the printer in the format required by the printer [1]. Preflight warns about missing fonts, verifies linked graphics, colors, page size and page setup. [14, 17]. It also verifies annotations, color space, compression, document properties (orientation, pages, producer, creator, version, etc.), font, forms, halftone, image, miscellaneous, rendering and transparency [1].

Preflight tests the validity and completeness of a prepared document using predetermined rules. The set of rules that are used to check the files is called preflight profile. The preflight profile includes rules about the properties of the job such as page, text, color, image, font formatting and more. It is used to check the job content against certain characteristics that are related to a specific type of job.

CHAPTER 3
ARQUITECTURE

3.1 Metadata in Digital Publishing

Each stage of a digital publishing workflow may produce valuable information that can be used to annotate the different processes of a digital printing job such as scheduling and production tracking. These annotations will generate metadata which provides fundamental information that can enable collaboration among those elements involved in the printing process –from creation to delivery–by supplying information about fonts, layout, geometry, color management, and others. Metadata also facilitates retrieval and reuse of information resources, and provides valuable management information for analysis and optimization of jobs, so that multiple jobs can progress through different stages of the workflow. Our contribution in this area is a set of methods created to incorporate, track and analyze metadata generated at each stage of a digital workflow.

First, we have created a pre-press ontology processes (refer to Section 3.3), whose terms and relationships were already defined explicitly in PDF and JDF (Job Definition Format) [15]. The ontology was focused on technical aspects of the processes and subsets were chosen of these languages accordingly. Other sources for the ontology design were digital publishing experts. The ontology was designed using Protegee, [18] section illustrates 3.3 , a free, open source ontology editor and knowledge based framework. Figure 3–7 shows a screenshot of the ontology

development in Protegee.

Java-based tools, such as Jena [19] and Jess [20] were evaluated for metadata analysis and decision-making support. Jena provides the mechanisms for parsing OWL (Web Ontology Language) [21] and RDF (Resource Definition Framework) [22], and provides basic inference models. Jess, an Expert System Shell, supports various knowledge representation schemes, and provides a variety of inference models and strategies that will be assessed according to the type of problems and decisions needed for workflow management in a digital publishing environment. It is expected that Jena and Jess will support decisions that are beyond the reach of business rule systems available in some workflow engines.

3.1.1 Intent and Registration

The Intent is one of the processes where further information is gathered from the client and the print jobs the client is submitting. The metadata from this process shown in table 3–2 and 3–1 captures relevant information about the expected quality of the job, the priority, the color systems, changes in the documents, type of job, and more. This process is essential for our system since each job is different, and its specifications will specify how the document will be treated in the Automated Workflow System. This intent process describes the jobs focusing on the capabilities of the service provider. In addition, it provides critical information about how the file was created, what is sent to the jobs, formats, requirements, and others. In order for a print job to be executed, all digital files must be delivered and managed properly,i.e;

- Special Format jobs require a certain grade of quality, paper, or finishing.

Table 3–1: Register

Key	*Values*
IDJob	A unique number is usually generated by customer service or the accounting department. This value is used to determine the status of the jobs, and its production flow.
IDCustomer	Customer Identification information, which serves to provide information about previous works already developed in the print shop facility.
ApprovalPerson	One who can approve changes in the documents.
TechSupport	Contact for technical aspects of the documents.
SubmissionDate	Time and date when the job was submitted.

– Font Usage - Fonts can be changed or not.

– All components are available, e.g. photos, special logos, etc.

– Hard Copy if provided by the client.

– Final Quantity

– Paper Type

– Finishing requirements

– Delivery Date

3.1.2 PDF

A PDF document may include information such as the document's title, author, creation and modification dates. The document's information is part of the metadata defined in the PDF file. Also, this metadata is used to assist in cataloguing and searching documents by external systems. The metadata of the PDF file can

Table 3–2: Intent

Key	*Values*
Required	This can establish how much time the process of production would take. This time is obtained from previous work, and previous knowledge from the customer. This time can be acquired from the metadata of previous work recorded in the Resource Description Framework format, explained in**??**. This information can be helpful for job administration, scheduling, billing and delivery.
PrintQuality	It establishes the quality expected from the client. This can be a numerical value in scale, where the errors that can filter through the system are determined.
JobDescription	This includes textual description of the print job, and other specifications. This information is defined in section 3.1.1.
PDFFile	Name of the file, and where it is located. This File may produce important metadata defined in 3.1.2.
PrintVersion	Different versions of the document can be received by the printshop. Eg. PCL,PDF,PS.
Material	This includes the inventory of the intended output print job. It Has all job elements, fonts, graphics, images with proper resolution, backup of the original files and special software needed to open any elements. This is important to determine if the job can be processed as it is, or if further in-house development is needed.
ChangesFont	Identifies if the client accepts any changes of font, in case this is not provided in the job.
PriorityCLient	Identifies if the client has a certain priority in the job process.This provides metadata for other processes involved in DP.
JobError	It is related to the PrintQuality.
Proof	List of deliverables, as proof of products in the production process.

be added or changed by users or plug-ins[17]. Metadata can be stored in a PDF document as the following methods:

- *Document Information Dictionary* associated with the documents. This metadata of the PDF file itself, where any entry whose value is not known, should be omitted, rather than included with an empty string. Plug-ins and software that manage PDF are able to search the content from the document information dictionary. This facilitates the browsing and editing of files. This metadata is shown in Table 3–4

- *Metadata stream* associated with the document or a component of the document. For proof of concept of the investigation we decided the use of metadata streams. They are more appropriate for the use of PDF-based workflow, where metadata of artworks and other components may be integrated. This metadata streams provide a standard way of preserving the metadata of these art components that can lead us to its examination. The PDF application should be able to manage this metadata within the document itself. Since it serves both methods, this stream lets the tools examine, catalog and classify documents. These tools should be able to understand the self-contained description of the document, even if the tool does not read or render the PDF itself.

3.1.3 XMP

This metadata is represented in XML(Extensible Markup Language format). This information will be visible as plain text to tools that do not read or render PDF[17]. This metadata is described and defined as part of the XML framework Extensible

Table 3–3: PDF Metadata

Key	*Values*
Titles	The Document Title.
Author	The name of the person who created the document.
Subject	The subject of the document.
Keywords	Keywords associated with the document
Creator	If the document was converted to PDF from another format, it refers to the name of the application(for example, Adobe frameMaker) that created the original document from which it was converted.
Producer	If the document was converted to PDF from another format, it applies to the name of the application (for example, Acrobat Distiller) that converted it into PDF.
CreationDate	The date and time the document was created, in human-readable form.
ModDate	The date and time the document was most recently modified, in human-readable form.
Trapped	A name object indicating whether the document has been modified to include trapping information True- The document has been fully trapped, no further trapping is needed. False- The document has not yet been trapped; any desired trapping must still be done.

Metadata Platform. This framework provides a way to use XML to represent metadata describing documents and their components, and is intended to be adopted in most of applications that process PDF. Figure 3–1 shows an example of an XMP file containing the information of a Digital Camera Photo Smart.

3.1.4 Preflight

In the digital publishing automated workflow, all parties must have in depth information and knowledge of the tools, specifications, and technology used to create a job. Mostly, customers must work with print shops to understand the technical considerations that need to be accessed in order to print properly. Such considerations include color expectations, price, turnaround, time, finishing and technical limitations. These files must be checked before they are sent to the output devices, and the metadata shown in Figure 3–4 will help to provide a high quality print

```
<rdf:Description rdf:about=''  xmlns:pdf='http://ns.adobe.com/pdf/1.3/'>
</rdf:Description>

<rdf:Description rdf:about=''  xmlns:tiff='http://ns.adobe.com/tiff/1.0/'>

 <tiff:Make>Hewlett-Packard</tiff:Make>

 <tiff:Model>HP PhotoSmart R717 (V01.00) </tiff:Model>

 <tiff:Orientation>1</tiff:Orientation>

 <tiff:XResolution>300/1</tiff:XResolution>

 <tiff:YResolution>300/1</tiff:YResolution>

 <tiff:ResolutionUnit>2</tiff:ResolutionUnit>

 <tiff:YCbCrPositioning>1</tiff:YCbCrPositioning>

</rdf:Description>
```

Figure 3–1: XMP Metadata HP Digital Cam

job. Information such as price, quality, speed of job, manufacturing path or control flow, and errors, are part of the metadata for digital publishing.

3.1.5 Preflight Results

The preflight results can be point to some areas inside the document, where errors and warnings can be found as shown in Figure 3–2. This way, the workflow manager can send portions of the job for further evaluation and not the entire job. This helps in reducing processing time of the job at the time of the RIPPING process, and results in a more efficient use of resources. This metadata, in combination with the Intent metadata, precede the artifact recognition analysis.

Currently, there are other processes which help develop a PDF file to be printed correctly. Some of the processes are explained in this section and are involved

Table 3–4: Preflight Profile

Key	*Values*
Preflight Profile	
Document	
PDF	A PDF document requires at least Acrobat 4.0, 5.0, 6.0, 7.0.
Encrypted	Whether the document is encrypted or not Special keys are needed
Document Damaged	
Page Size	equal, less than, more than, unequal
Orientation	Page Size may vary from page to page
measurement	Inches,centimeters, picas , points millimeters
Images	
ResolutionColorLow	Lower than X ppi.
ResolutionColorHigh	High than X ppi.
ResolutionGrayLow	Lower than X ppi.
ResolutionGrayHigh	High than X ppi.
ResolutionBitmapLow	Resolution of bitmap images is lower than than X ppi.
ResolutionBitmapHigher	Resolution of bitmap images is higher than than X ppi.
ImagesCompressed	
ImagesLossyComp	Images use lossy compression.

Errors & Warnings

	Severity	Description
✘	Error	Object overlaps page safe type zone (15x)
✘	Error	Effective resolution of single-bit black & white image is less than 550 dpi (11x)
★	Fixed	Changed page box layout in conformity with the press layout specifications
✘	Caution	Compression ratio of image is more than 10.0 (4x)
★	Caution	Not all pages in the document have the same size
✘	Fixed	Font Helvetica has been embedded (11x)
✘	Fixed	Font TimesNewRoman,Bold has been embedded
✘	Fixed	Font TimesNewRoman has been embedded (6x)
✘	Fixed	Font TimesNewRoman,Italic has been embedded
✘	Fixed	Black text is set to overprint
✘	Caution	X and Y scaling of image differs 0 % (4x)

Figure 3–2: PDF Preflight Result

in the ontology metadata. With these results we are able to eliminate the most common errors in file preparation,eg.;

- Fonts not embedded
- Wrong Color space
- Images missing
- Overprint/trap issues: There are several features that make reviewing problems identified in the Preflight Results Report easier.

  Hits per Rule: This feature allows the user to limit the hits per each rule. For instance, if the user forgets to replace 10 low-resolution placer files with high-resolution images, most likely the user will not need to be informed of the infraction 10 times. In addition, if the user intends to print a copy of the report, perhaps for the person who will make the corrections, it is best to limit the reiterative information. To generate a report, Adobe Acrobat Professional is used. The Preflight Report can be exported to ASCII text, XML, or PDF as shown in Figure 3–3.

3.2 Print Settings

As shown in figure 3–4, these settings are transfered to the device. These settings are configured based on the intent process.

3.2.1 Color Management

The primary goal of color management is to obtain a good match across color devices. Some Color management settings are used in the Digital Publishing Workflow, and form part of the metadata. These settings are shown in Figure 3–5.

3.2.2 PostScript Description Files

PostScript Printer Description(PPD) files contain information about the vendors, which include a set of capabilities and features in certain devices. These PPD files

Image Information

	Type	Color Space	Physical Res.	Effect.Res.(dpi)	Page	Angle (degrees)	Skew	Flipped	Custom Transfer	Custom Halftone	Custom BG	Custom UCR
✘	Mask	Gray	2368x3248	300.2x300.2	1	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	2	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	3	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	4	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	5	0.0	-	-	-	-	-	-
✘	GrayScale	Gray	2251x1277	300.1x300.5	5	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	6	0.0	-	-	-	-	-	-
✘	GrayScale	Gray	2191x987	300.5x301.1	6	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	7	0.0	-	-	-	-	-	-
✘	GrayScale	Gray	2162x1230	300.5x300.2	7	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	8	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	9	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	10	0.0	-	-	-	-	-	-
✘	Mask	Gray	2368x3248	300.2x300.2	11	0.0	-	-	-	-	-	-
✘	GrayScale	Gray	2182x958	300.4x301.2	11	0.0	-	-	-	-	-	-

Figure 3–3: Preflight Image Information

```
PRINT SETTINGS

PPD: HP LaserJet 5Si/5Si MX PS, (seaprint1-ntWiley)
Printing To: Printer
Number of Copies: 1
Reader Spreads: No
Even/Odd Pages: Both
Pages: All
Proof: No
Tiling: None
Scale: 100%, 100%
Page Position: Upper Left
Printer's Marks: None
Bleed: 0 in, 0 in, 0 in, 0 in
Color: Composite, Colors in Black.
Trapping Mode: None
Send Image Data: Optimized Subsampling
OPI/DCS Image Replacement: No
Page Size: Letter - Half
Paper Dimensions: 8.5 in x 11 in
Orientation: Portrait
Negative: No
Flip Mode: Off
```

Figure 3–4: Print Settings

```
Color Management
 Document Profile: U.S. Web Coated (SWOP) v2
 Color Handling: Let InDesign Determine Colors
 Printer Profile: Document CMYK - U.S. Web Coated (SWOP) v2
 Preserve CMYK Numbers: On
 Proof Profile: N/A
 Simulate Paper Color: N/A
```

Figure 3–5: Example Color Mangement Metadata

serve also for validation purposes to a certain Output Printer device, an example is shown in 3–6. These specifications include:

- Input paper trays
- Page Size definitions
- Print areas for each page size
- Output Paper trays
- Duplexing (double sided printing)
- Default font
- Screening Definition
- Default screen angles
- Black and White or Color
- Halftone Screening functions
- Default transfer functions
- Resolutions Available
- Memory Configuration

3.2.3 Resources

Not only are processes considered in the metadata, it is also need to identify which resources are available, or better for a certain job. These resources can include information such as Printer Technical Specifications, since each job is different and has different requirements. This information is useful for the Decisional Support System explained further in Chapter 4.

```
*% ================================
*% Basic Device Capabilities
*%================================
*LanguageLevel:         "2"
*ColorDevice:           True
*DefaultColorSpace:     CMYK
*TTRasterizer:          Type42
*FileSystem:            False
*Throughput:            "10"
\\
```

Figure 3–6: Example of a PostScript Printer Description Files

3.3 Ontology Development

3.3.1 Use of Protegee

Protegee is a tool for ontology editing and knowledge acquisition. Protegee has been used for projects ranging from modeling cancer-protocol guidelines, to modeling nuclear-power stations and now for Digital Publishing metadata presented in this thesis work. Protegee is aimed at making it easier for knowledge engineers and domain experts to perform knowledge-management tasks. One of the major advantages of Protegee architecture is that the system is constructed in an open source, modular fashion. Its component-based architecture enables system builders to add new functionality to Protegee by creating appropriate plug-ins such as, support for alternative storage formats, and domain-specific user-interface components. From Protegee, it is possible to export an ontology to other knowledge-representation systems, such as RDF, OIL and DAML. This format will help us in the distribution of the ontology, and make the metadata more useful. Figure 3–7, shows part of the ontology being described in Protegee. In this figure, it can be seen the ontology editor for editing the concept of Font, of the Digital Publishing Ontology. This shows how Font is represented as a subclass of the class Resources. This editor also provides the roles of concrete or abstracts, but that depends on whether we create direct instances of the class or not.

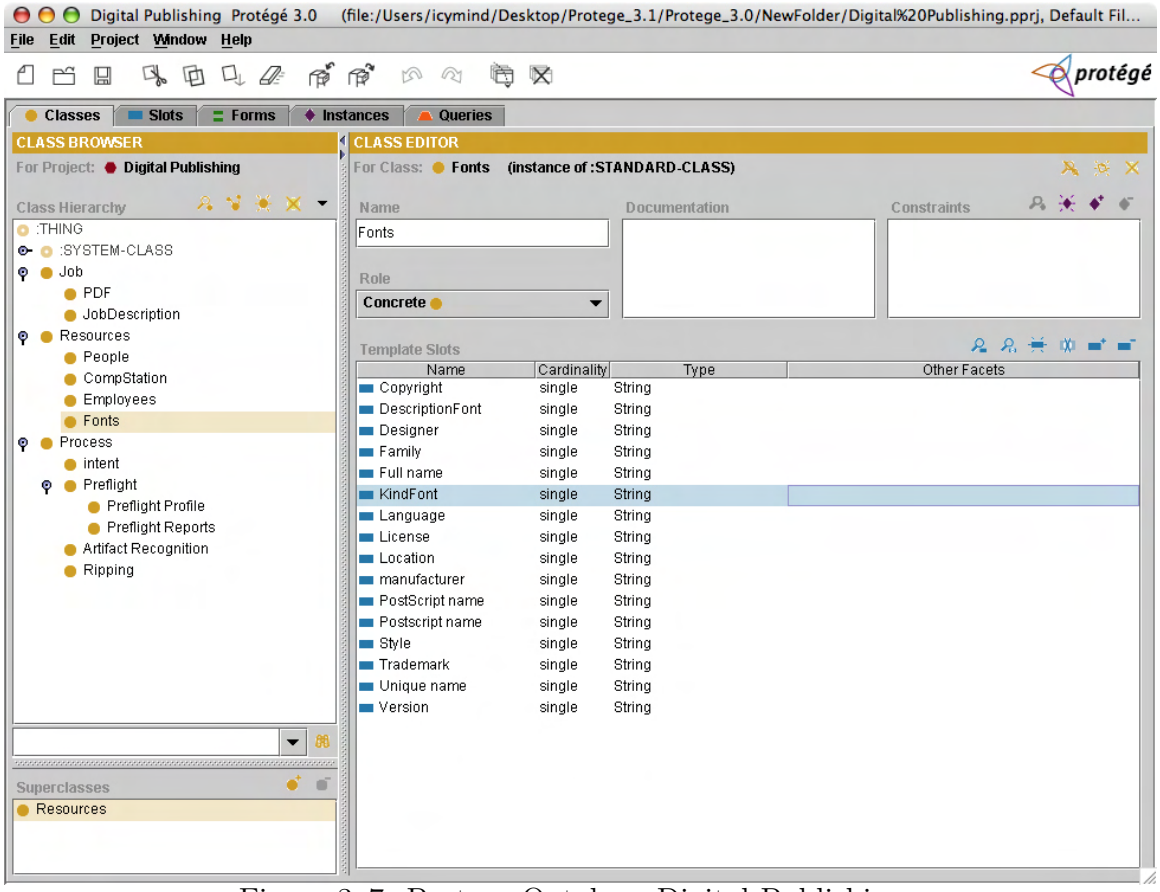

Figure 3–7: Protege Ontology Digital Publishing

Methodology

An ontology is a logical theory that expresses part of a conceptualization model. It represents an intentional semantic structure that illustrates implicit rules constraining the structure of a piece of reality [23]. The development of an ontology is essential for a knowledge-based system since every knowledge model has an ontological representation of the underlying conceptualization and logical theory [24]. An ontology can be designed in a top-down or bottom-up approach [25] The development of an ontology for a large knowledge based system can take years [26].

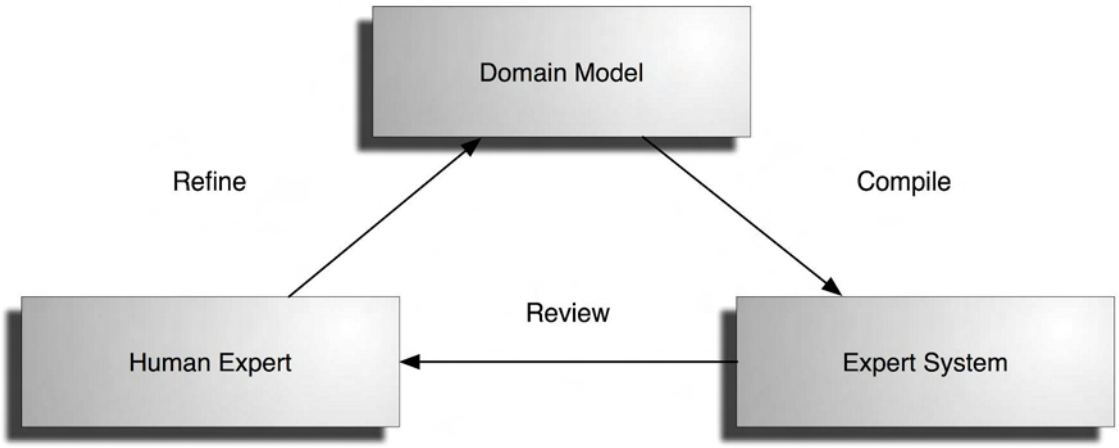

Figure 3–8: Knwoledge Acquistion

To build this ontology, a series of steps need to be performed:

- **Identify the purpose and the scope** - why is the ontology being built, and its intended use (shared, used, or reused as part of Knowledge base). In this step, relevant terms of the domain are identified. As for digital publishing, the general processes (intent, preflight, proofing and ripping) are established.

- **Build the Ontology** - Ontology captures the *Identification* of the key concepts and their relationships in the domain of interest; the *Production* of precise unambiguous text definitions for such concepts and relationships, and *Coding*

explicitly representing in a formal language, integrating in existing ontologies the knowledge acquired in the previous step. During the capture and coding processes, there is the question of how and whether to use ontologies that already exist; for the best of our knowledge digital publishing does not have a standard or official ontology. Part of the investigation is to find out how the development of a Digital Publishing ontology helps the knowledge base system in making decisions in the workflow system.

- **Evaluation** - here the authors adopt the definition of evaluation to make a technical judgment of an ontology, their associated software environment, and documentation with respect to a frame of reference. The frame of reference may require specifications, competency questions, or the practice in applications.

- **Documentation** - recommend guidelines for documenting ontologies, possibly differing according to the type and purpose of the ontology. A guideline example is to locate similar definitions together and create naming conventions such as using upper or lowercase letters to name the terms.

This methodology is based on experiences of building the enterprise ontologies by Ushold and King (1995) [10]. An ontology adds value to the Digital Publishing business, since it allows us to reuse and share the knowledge components of a domain. Problem solving methods can be realized, making dynamic changes and inferences on the digital publishing workflow. An ontology creates a contextual terminology in that it can be used for information sharing and exchange, and gathering new implicit knowledge [27].

For this ontology to communicate with workflow systems, it is need to establish a service to implement a common web services definition language (WSDL) interface for the Web Services Information [28][29] . This communication can be carried out over HTTP for Web context, but in other context protocols such as Remote Method Invocation (RMI) or the Simple Object Access Protocol (SOAP) may be more suitable.

Using the Sesame version of RQL [30] **??** which features better compliance to W3C specifications, let us accomplish the deployment of the ontology in a practical environment, including support for optional domain and range restrictions. The query module allows us to create queries in different levels. This query module will be explained in 3.4. Query as :

- **Syntatic Level** through simple XML documents.

- **Structure Level** consisting of a set of triples.

- **Semantic Level** through one or more graphs with partially predefined semantics.

The Repository Architecture serves as a simple repository, where the knowledge base system resides. The request router will keep the communication from a workflow system and/or expert system. The module administrator can act as knowledge engineer to add knowledge to the system. The Export module can let us create a single file with all the information residing in the repository system.

The knowledge-based system presented in the next chapter, stores and represents knowledge within an organization. The use of a rule-based system that will be

developed in JESS [20] adds the capability to manage this knowledge base and reach conclusions that help the workflow engine or the operator to make formal decisions on digital publishing processes. Through the use of knowledge based systems, as shown in figure **??**, more capabilities can be added to the decision support system of the digital publishing workflow system. The knowledge base can be applied to other steps of the workflow system to discover new information that can be acquired with it. The information in the ontology and the knowledge-based system can represent domain knowledge that can be re-used for any processes in the workflow management to make better decisions about job routing.

RDF is the language used to represent knowledge about the domain of digital publishing. There is an almost one-to-one correspondence between the classes and objects in RDF and JESS. A mapping between the RDF representation and JESS can be established so that the inference engine in JESS can be used with the ontology.

3.3.2 Class Definitions

Many approaches exist to define the classes and subclasses in ontology. For the digital publishing application it was established a top-down development process (Uschold) where it began with the definition of general concepts in the domain, reaching down to a more detailed specialization of the concepts . In figure 3–9 it is shown a simple graph of the ontology, but further details need to be added which will be explained in section 3.4.

3.4 RDF

RDF is a description for data models rather than a description of a specific data vocabulary. RDF (Resource Description Framework) is a way to make relational

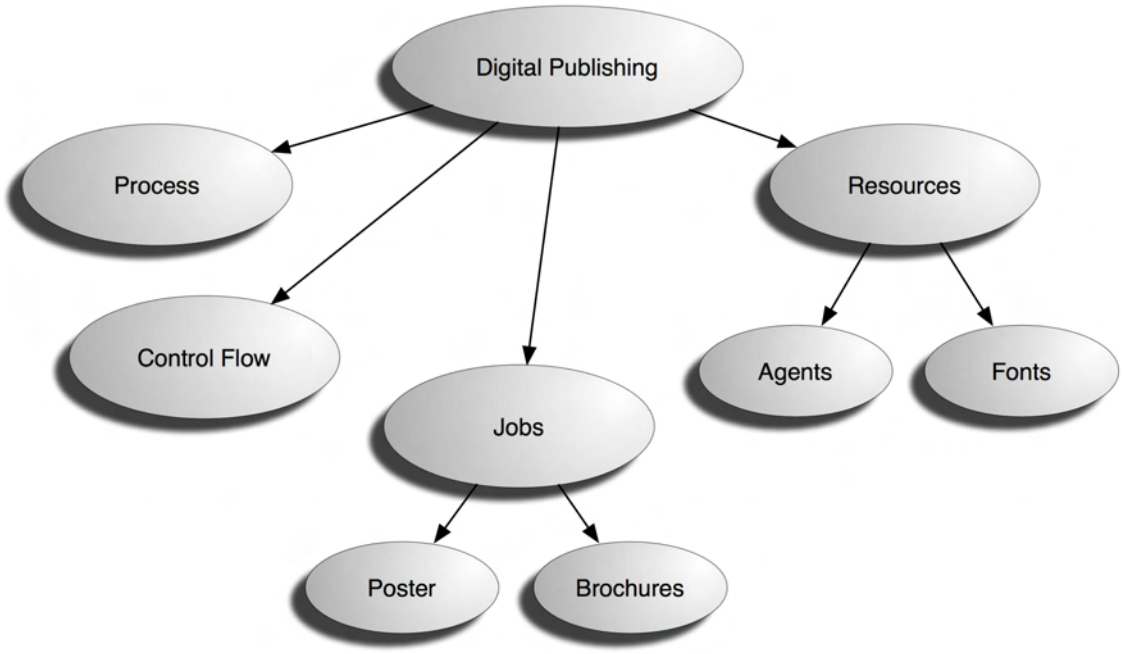

Figure 3–9: General Ontology Digital Publishing

data for the web, that allows the use of structured and semi-structured data to be mixed, exported and shared across different applications. RDF data describes documents using XML schemas to define a document. Nowadays, RDF does not limit to documents, and is used for various commercial products for better search, catalogs, information retrieval, semantic web and other research areas. RDF and OWL schemas are also known as ontology applications where XML provides interoperability within an application, and across applications.

RDF is a way of recording information about resources. As serialized using XML, is a way of recording information about a specific business domain using a set of elements defined within the rules of the RDF data model/graph and the constraints of the RDF syntax, vocabulary and semantics. Since RDF helps for domain specific business, it provides an ideal data environment for the Digital Publishing workflow. The next section will present the structure of the metadata for digital publishing in RDF.

3.4.1 RDF Structure

Metadata itself is used for part of a document. We use RDF to create metadata to capture the data about the external use of each process in the digital publishing management work, like the intent already explained in section 3.1.1, including, author, dates, requirements, type of job. The first parts of the documents specify that it is an RDF document–these are the root elements and are required in any RDF validated file. The documents contain one or more "descriptions" of the resources. A description is a set of statements about resources, which in this case, describes many aspects of Digital publishing.

3.4.2 RDF Data Model and the RDF graph

RDF itself provides a model that is often called a "triple" because it has three parts. The RDF triple describes terms of resource properties from the knowledge representations, which are described as grammatical parts subject, predicate and object. An example of the RDF triple Figure 3–10 .

RDF identifies things using Web like identifiers known as Uniformed Resource Indetifiers(URIs), and describes resources with properties and property values, as explained below:

- A Resource is anything that can have a URI, such as "ArialFont", or any type of data described by RDF; they are also referred to as URI(Uniform Resource Identifiers)

- A Property defines attributes or relations used to describe a resource that has a name, such as "manufacturer" or "location.

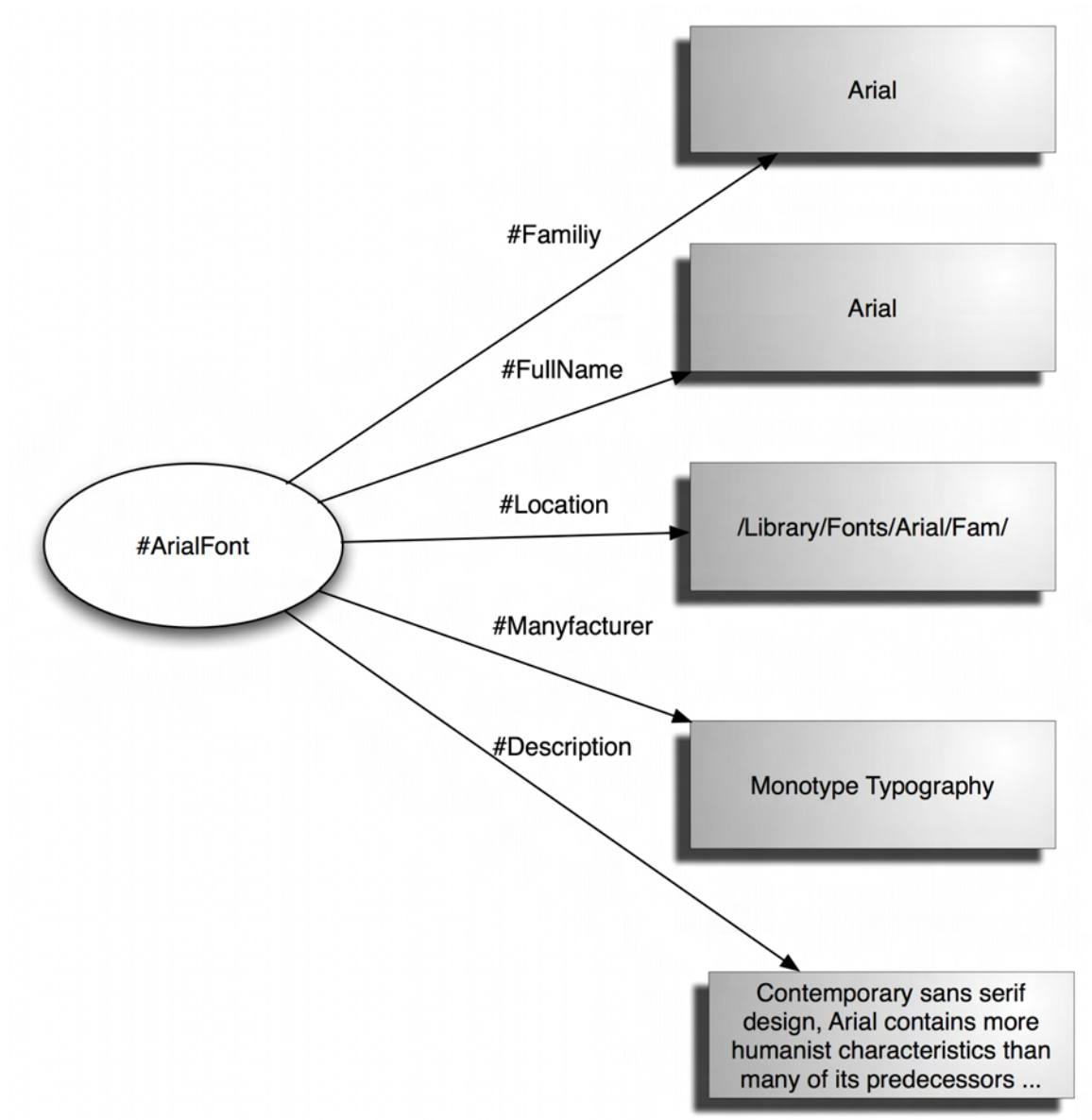

Figure 3–10: RDF graph for a Font Instance

  - A Property value, such as "Monotype Typography" or "/Library/Fonts/Arial/fam" (note that a property value can be another resource), anything that assigns a value to a property of a certain resource.

The combination of resource, property, value, is also known as an RDF statement where in this case it was established that ArialFont is a property of Monotype Typography and is located in /Library/Fonts/Arial.

From a grammatical point of view, a resource description in RDF, is composed of the following components:

- subject of the statement: ArialFont
- predicate: Property
- Object: Monotype Typography

We chose the use of RDF for digital publishing, as a tool to develop the metadata for Digital Publishing Features such as :

  - Interoperability of data
  - Machine understandable semantics
  - Better precision in resources discovery
  - Rules to allow for decentralized extensions
  - Descriptive rather than prescriptive(Contrast XML only syntax)
  - RDF is designed for merging data.

Further development will enable RDF to also provide a uniform query capability for resource discovery, processing rules for automated decision-making and a Web resource language for retrieving metadata from third parties. The significant benefit that RDF brings about is that it will allow the resource description communities to primarily focus on the issues of semantics rather than the syntax and structure of metadata. The contents of the framework will be determined by the stakeholder communities - independently developed and maintained. RDF also

<rdf_:Fonts rdf:about="&rdf_;Arial Font"
rdf_:Family="Arial"
rdf_:Full_name="Arial"
rdf_:KindFont="TrueType"
rdf_:Location="/Library/Fonts/Arial"
rdf_:Postscript_name="ArialMT"
rdf_:manufacturer="Monotype Typography"
rdfs:label="ArialFont">

<rdf_:Copyright>Typeface ¬© The Monotype Corporation plc. Data ¬© The Monotype Corporation plc/Type Solutions Inc. 1990-1992. All Rights Reserved
</rdf_:Copyright>
<rdf_:DescriptionFont>Contemporary sans serif design, Arial contains more humanist characteristics than many of its predecessors and as such is more in tune with the mood of the last decades of the twentieth century. The overall treatment of curves is softer and fuller than in most industrial style sans serif faces. Terminal strokes are cut on the diagonal which helps to give the face a less mechanical appearance. Arial is an extremely versatile family of typefaces which can be used with equal success for text setting in reports, presentations, magazines etc, and for display use in newspapers, advertising and promotions.</rdf_:DescriptionFont>
<rdf_:Designer>Monotype Type Drawing Office - Robin Nicholas, Patricia Saunders 1982</rdf_:Designer>
<rdf_:Language>English, French, German, Spanish, Italian, Swedish, Danish, Finnish, Portuguese</rdf_:Language>
<rdf_:License>NOTIFICATION OF LICENSE AGREEMENT This typeface is the property of Monotype Typography and its use by you is covered under the terms of a license agreement. You have obtained this typeface software either directly from Monotype or together with software distributed by one of Monotype√ís licensees. This software is a valuable asset of Monotype. Unless you have entered into a specific license agreement granting you additional rights, your use of this software is limited to your workstation for your own publishing use. You may not copy or distribute this software. If you have any question concerning your rights you should review the license agreement you received with the software or contact Monotype for a copy of the license agreement. Monotype can be contacted at: USA - (847) 718-0400 UK - 01144 01737 765959 http://www.monotype.com</rdf_:License>
</rdf_:Fonts>

Figure 3–11: RDF file Font Instance

allows for re-use, extendibility and refinement of the established resource description standards since these will be available in machine-readable form. The new Resource Description Framework specification is an exciting new challenge to the resource description communities as there now will be a standard mechanism for the global exchange of metadata and their schemas. The consistent use of metadata and application of metadata schemas means that semantic interoperability will be preserved, hence, significantly improving the deployment ability of advanced Web applications.

Making this information available to computers in order to enhance their usefulness, was the driving vision that created the Semantic Web project. Most traditional metadata approaches take the view of meta-data as being mostly a digital indexing scheme to use in cataloging and digital libraries, allow reasoning and inference capabilities to be added to the pure descriptions. In its simplest form, this includes stating facts such as Arial font is a Resource in the creation of a Document but extends the deduction of complicated relationships. This is an important feature to allow intelligent agents and other software to not only passively swallow descriptions, but to act on them as well. The Semantic Web is a web-technology that lives on top of the existing web, by adding machine-readable information without modifying the existing Web. It is designed to be globally distributed for scalability and flexibility.

3.4.3 Describing this Metadata with RDF

It is not immediately obvious that the simple statement model of RDF can be used to make the Semantic Web a reality. The most fundamental benefit of RDF compared to other meta-data approaches is that using RDF, describe anything about

anything. Anyone can make RDF statements about any identifiable resource. Using RDF, the problems of extending meta-data and combining meta-data from different formats and from different schemas disappear, as RDF does not use closed documents.

3.4.4 RDF Schema

The RDF schema can provide a primitive way to model an ontology. With this model, the data can have a perfect balance between expressiveness and reasoning for the needs of the application. Resources may be divided into groups called classes, already explained in section 3.3.2The members of a class are known as instances–as the example of an RDF Font instance shown in figure 3–11.In RDF it can be specified other classes such as resources, sometimes called *RDF URI References* and may be described as RDF properties.

RDF Schema is a language for describing vocabularies in RDF. RDF Schema is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources. RDF Schema vocabulary descriptions are written in RDF using the terms described in the RDF Schema specification. These resources are used to determine characteristics of other resources, such as the domain and range of properties.

We use the RDF schema to provide a logical way to describe the ontology for Digital Publishing in a data system where other applications can make information useful.

3.5 Knowledge Sharing

3.5.1 RDF query

Currently, most of the inference systems for RDF are mainly devoted to querying information about RDF ontologies as if they were deductive databases, but some languages like RDQL and RQL can query at a higher level, something that is difficult to achieve for simple database queries.

Table 3–5: Important use of RDF in Digital Publishing Ontology

Feature	*Short Description*
describe	Since a resource can have uses outside the domain foreseen by the author, any given description (meta-data instance) is bound to be incomplete. Because of the distributed nature of RDF, a description can be expanded, or new descriptions can be applied
certify	There is no reason why only big organizations should be able to certify content - individuals may want to certify a certain content as a quality learning resource that is well suited for specific learning tasks. How to handle this kind of certification will be an important part of the Semantic Web and other machine-learning technology.
annotate	Everything that has an identifier can be annotated. There are already attempts in this direction: Annotea8 is a project where annotations are created locally or on a server in RDF format. The annotations apply to HTML or XML documents and are automatically fetched and incorporated into web pages via a special feature in the experimental browser Amaya9.
extend	Structured content (typically in XML format) will become common. Successive editing can be done via special RDF-schemas allowing private, group consensus or author-specific versions of a common base document. The versioning history will be a tree with known and unknown branches which can be traversed with the help of the next generation versioning tools.
reuse	RDF is application independent. As the meta-data is expressed in a standard format independent of more advanced schemas that are used, even simplistic applications can understand parts of large RDF descriptions. If more advanced processing software is available (such as logic engines), more advanced treatment of the RDF descriptions is possible.

RDF metadata and the RDF schema can be seen in different levels.

- Syntatic Level
- Structure Level
- Semantic Level

Syntatic level: RDF is based on an XML notation. We can query the RDF using normal XML query languages like XQuery. However, this kind of query is not suitable for a better management of metadata in digital publishing. This syntactic level is useful only in searches of the XML document itself.

Structure level: this especially in the format of RDF, which consist of a set of triples as already explained in section **??**. Few query languages have been implemented, but the problem of these query languages is that they interpret the RDF only as a set of triples, including those elements that are included in the RDFS. The lack of query is that some information can be implied in the RDFS.

Semantic level: To take advantage of the Ontology specified in digital publishing, we will be querying the full knowledge that an RDFS description contains. It is needed to compute and store the closure of the given graphs as a basis for querying and letting the query processor infer new statements as needed.

Many languages like RQL **????** can use a declarative query language that explicitly captures these semantics in the language design itself. RQL is based on syntax OQL (Cattel 200) and like OQL, it allows the use of queries of classes, properties or instances. One implementation of this is Sesame (ref). Using their architecture for query languages we deposit the Digital Publishing Ontology to make all these kinds of queries.

3.6 Chapter Conclusion

Digital publishing uses a lot of data. Providing this framework and the capabilities to add knowledge to the ontology, publishing can be an easier job. We can provide the metadata involved in the work of the production environment to describe the

importance of the metadata, and make the information useful for other applications. This metadata provides all details of the client and the print-shops, such as job estimates, work order, customers information, and others. Also, this metadata helps the workflow system by providing the information needed for certain operations such as; establishing names, titles, individuals who work, and client approval.

This metadata makes the workers and the resources that are available in a certain print-job more useful. With these approaches a better job is expected by helping to reduce, time, money, and materials. Providing this framework definition, other applications can be developed, such as an expert system that is shown in Chapter 4 as an example of an Application of how useful this metadata is in a certain process.

CHAPTER 4
EXPERT SYSTEM

4.1 Expert System

Nowadays there is no effective way to determine if a document can be printed or not for a specific job in digital publishing. Many times, the preflight expert at the print shop has to design his/her own methods to determine the success of jobs through many steps inside the print shop. Once the preflight profiles of the print shop are defined the expert has to evaluate each job in order to assign the best profile for each preflight process. In addition, the expert reads the results of the preflight report, and with the previous knowledge about the print shop, he/she makes a decision as to whether the job can be printed or not.

By establishing the need for an easier way to determine the results of each job based on of the job metadata, we are proposing an expert system tool that suggests and supports the decision, based on information about the job from the process already explained in section 3.1.4

Our expert system is an analytical diagnosis tool that will infer if the document can be printed from the input data received from the user or the metadata collected in any step of the Digital Publishing Workflow. It is also a synthetic design system that will use constraints given by the user as part of the criteria to determine whether or not the document can be printed based on the knowledge of the print shop itself as shown in figure 4–1. It is also a synthetic planning system that will

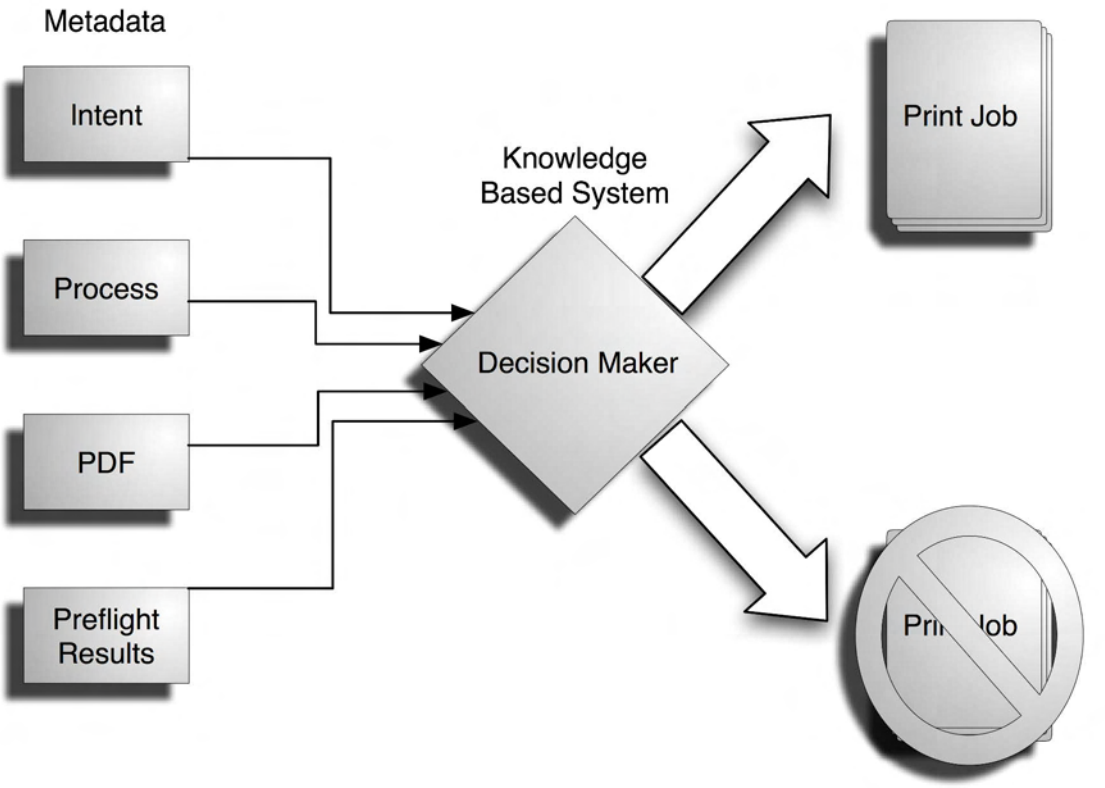

Figure 4–1: Decision Support Scenario

generate a sequence of actions, which will depend on the user feedback to make the right decision for the job submitted. It may be used for digital publishing, or desktop publishing jobs, and from printable documents to view-only documents.

4.1.1 Jess

Jess is a rule engine and scripting environment written entirely in Sun's Java language by Ernest Friedman-Hill from Sandia National Laboratories in Livermore, CA. Jess was originally inspired by the CLIPS expert system shell, but has grown into a complete, distinct, dynamic environment of its own. Using Jess, a Java engine expert system, an expert system can be built that has the capacity of "reasoning" by using knowledge supplied in the form of declarative rules. Jess is small, light, and one of the fastest rule engines available [20].

4.1.2 Inference Engine

The inference engine Jess executes rules in a given order to solve a certain problem. By doing this, it can get new facts and knowledge. In the Digital Publishing application an interaction with the system is possible through the workflow system, the user or any other system. The inference process is composed of the following:

- Detection. The Rules of the KB are evaluated by matching the working memory against the invocation conditions. Applicable rules are chosen and stored as a new set.
- Selection. It refers to set of applicable rules, where one is selected for the first execution. The selections strategy can be set by the inference engine e.g. (first rule in order, rules with the easiest evaluations, most used rules, most specific rules, most general rule, highest priority.)
- Applications. these actions lead to other operations or a conclusion, it activates the rule that is executed and selected. A conclusion can generate a new sub goal that modifies the working memory.

4.1.3 Methodology

The process of collecting information about the domain of preflight profile was based on books, experience and information from web sites. This information was used in the ruled-based system Jess [4] to represent this knowledge.

Looking at the problem, we first established the requirements based on observations and then defined the problem that the system needed to solve. We learned the organizational principles of the field once we understood the need, of resources such as people, software, knowledge and other helpful information to add knowledge to the system. Then, we needed to know the scope, defined as the problem about the domain, where we needed to define the ambiguities and limits of human

understanding. That is why the ontology explained in section 3.3 is useful for the expert system development.

In this problem it can be observed how information differs from one source to another, since we get information overload, and find many conflicting resources.

4.1.4 User Modeling

The system needs to adapt to the user and a specific audience, to explain important aspects of the decision being made. The audience not only affects the explanation levels, but also influences explanation content. User modeling enables the system to construct internal representations of user knowledge, goals, and plans. By analyzing the user interaction with the system and by using this to generate the explanation queries, the system will provide the most relevant information possible.

This system is actually built for a specific group of users,in such a way that it can be modified and adapted to any user level. All these features contribute to expert system explanation module. To make matters more complicated, graphic designers are rarely certain of answers to any of the questions.

4.1.5 Architecture

The system has a modular design to improve its understanding. Modifiability is an important quality for software systems, because a large part of the costs associated with these systems are spent on modifications. The effort, and therefore cost, that is required for these modifications is largely determined by the system's software architecture. That is why this digital publishing DSS framework is developed to easily adapt to the need of each print shop.

General overview of the Expert System:

- Initialization of the Expert System

- Gathering metadata based on the type of job
  - Intent Module
  - PDF Metadata Module
  - Preflight Metadata Module
  - Preflight Results
- Recommendation of the Proper Solution

- Diagnostic for the preflight decision support

As any other system, we have a startup module that establishes the welcome procedure, asking the name of the user and relevant information for further presentation purpose. This is basically a welcome banner.

First, we establish the rules that fire the questions will be asked from the intent module. The purpose of this module is to ask questions to the user to collect information. The *deffacts* contain the potential questions that the system may ask, depending on the information gathered from the user. The first module is a set of rules to ask questions to the user to collect the data about each module. This process was explained in section 3.1.

The system does not ask questions blindly. The rule that asks each question will often need to match the existence or absence of some other fact, so that a question will not be asked unless certain conditions are met. In this module, the rules have two type of activation: one based on patterns of the answers already gathered from the user and a second one where the rules are activated by a call to reset. These are the first questions that the system will ask every time the system starts.

For our intent module the testing was done based on jobs received at the Digital Publishing facility at UPRM. We created a hypothetical case situation and an expected set of questions and results. This testing phase and additional experimental tests helped to create a more detailed system. In the Jess system, we used the facts present in all modules are monitored, to make sure all the expected facts were present and validated.

The recommendation maker module. This module needs to be developed, since the information depends on the industry itself and each print job works differently. Each rule verifies the information from the user and answers the facts and recommends one decision about the success of a certain job with their reasons. These rules encode the knowledge gathered from books, experts, Digital Publishing related site, and other resources.

Figure 4–1 shows the way the user and the expert system interact to choose the best solution for a certain job being submitted. First, the expert system gathers the data from the intent process; then it sends the file to the preflight tools, the report is analyzed and sent to the user. When the interview module finishes, the system recommends a profile according to the information provided by the user.

During the process of creating the expert systems, we worked on the designing and finally building a complete expert system. We gather information about the process of creating an expert system, that can be applied to applications such as diagnosis, troubleshooting, and other similar systems.

4.2 Knowledge Rules

Knowledge in Jess is represented as rules, as one of the primary methods of representing knowledge in this language. Rules decide the actions to take in a specific situation and are composed of an antecedent and a consequent. The antecedent

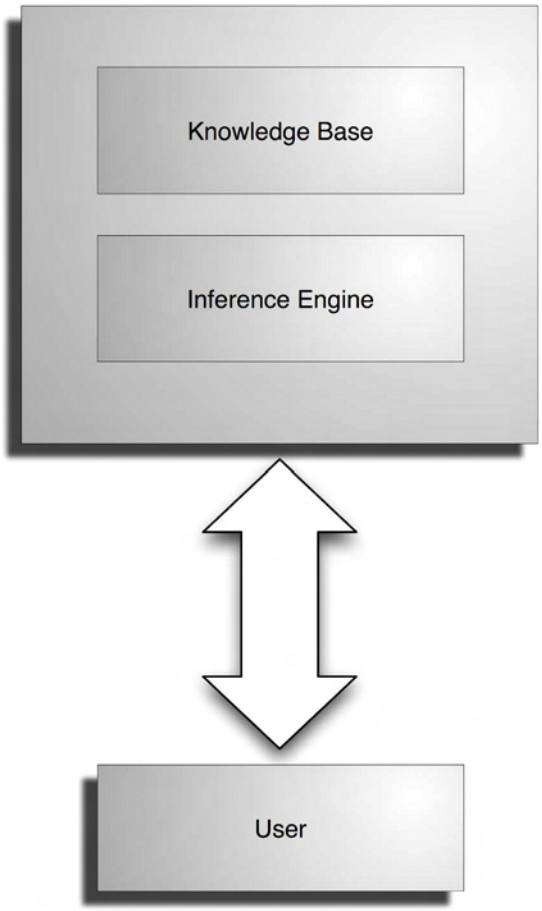

Figure 4–2: Expert System

of a rule is a set of conditions and the consequent of a rule is a set of actions to take if the conditions specified in the antecedent are satisfied (when the rule is applicable). The process to verify the conditions and fire the rule is carried out by the inference engine.

An important characteristic of the knowledge representation using Jess is that the inference engine is always checking the rules, and when the conditions of one rule are satisfied, it executes the actions. This process looks similar to the traditional IF-THEN instruction, used in procedural programming, but the substantial difference is that this type of conditions are only evaluated at a certain moment ,unlike in rules where the inference engine is checking the rule conditions all the time. In Figure **??** a rule is shown where we get basic information about a Black and White job .

```
  (defrule BWprintingspot
(answer (ident distribution)(text Hard-Copy))
(answer (ident fontChange )(text yes))
(answer (ident spotcolor)(text yes))
(answer (ident colorinfo)(text BW))
=>
(assert (recommendation (printjob BWSPOT)(explanation "This PDF targets two color
```

Figure 4–3: Example Rule in Digital Publishing Knowledge Base

4.2.1 Modules

Jess allows the expert system to control the execution and modular development of knowledge bases with the construction *defmodule*. Modules allow to group a set of constructions in order to maintain control over the restriction of the access from other modules; this means grouping rules in modules is possible to control their execution. These modules allow other modules to see only certain facts .

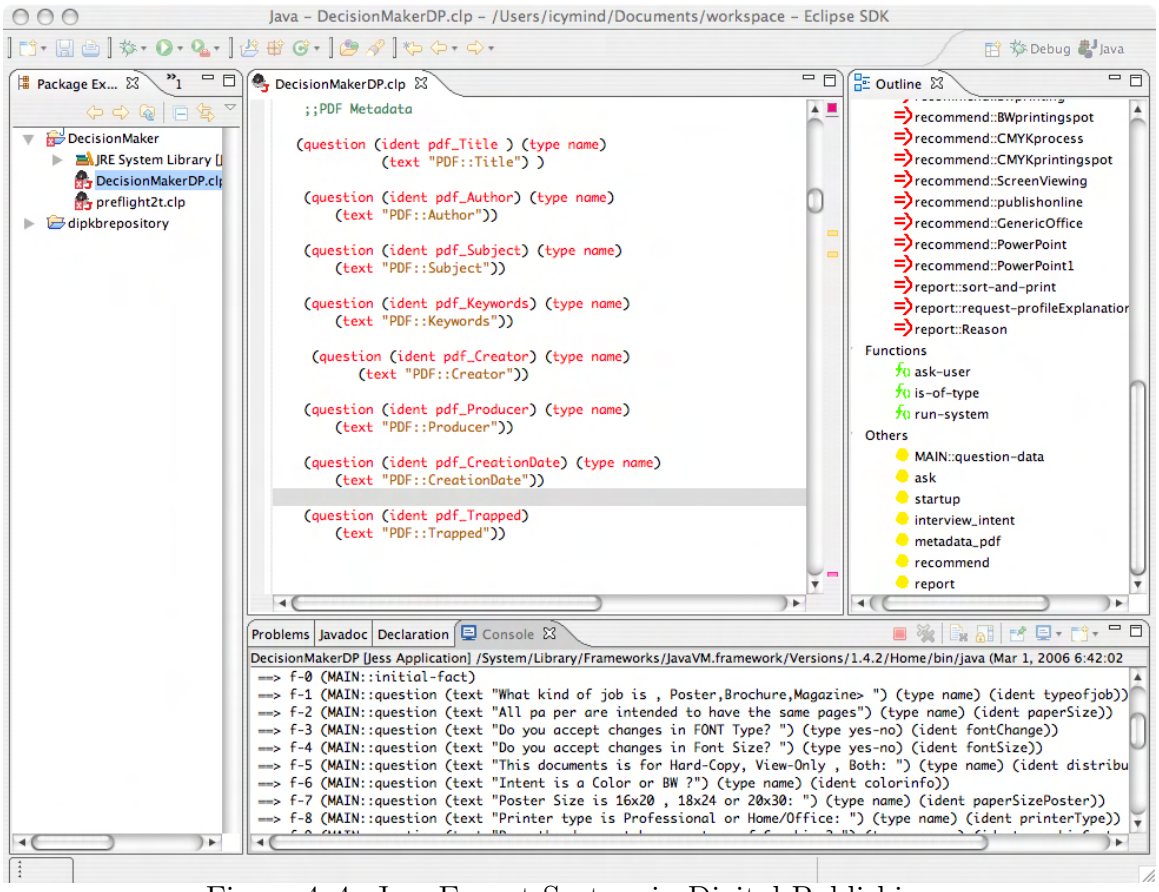

Figure 4–4: Jess Expert System in Digital Publishing

Defmodules allows knowledge to be partitioned. Every construct must be placed in a module. The programmer can explicitly control which constructs in a module are visible to other modules, and which constructs from other modules are visible to a module. The visibility of facts and instances between modules can be controlled in a similar manner. Modules can also be used to control the flow of rule execution. JESS provides support for the modular development and knowledge based execution with the defmodule construct. Jess modules allow a set of constructs to be grouped together in such way that explicit control can be maintained, thus restricting the access of the constructs by other modules. In this research, it can be seen the module declarations of a program that tries to get the metadata of each process defined. The program has four modules as shown in figure 4–4, each of them works on a specific task. Following the module declaration, it can be seen a intent module, but this belongs to the module of Gathering Metadata for Digital Publishing.

4.2.2 Decision of preflights

The system itself has to consider different variables, to make an informed decisions as follow:

- Quality and control of image acquisition and output. Low-resolution and/ or low color-depth digital images produce low quality print images. Extremely high-resolution and/or high color-depth images may exceed the capabilities of the output devices, and increase processing time substantially. There are several places in the process where these variables can be adjusted, including image capture/scanning, image creation or editing, page makeup and page output processing (RIP).
- Applications and versions Each application has its own intricacies that can affect the document and different applications/versions may treat content and images differently.

- Image formats. This Considers type, style, format, and compatibility with page layout applications, production process and file requirements.
- Fonts This considers type l, legalities, compliance with standards, compatibility with shop tools and electronic transfer methods.
- Platform and operating -system support. High Quality digital files can be huge and manipulation of these files requires powerful computers with substantial memory. Not all applications run on all operating systems. This, is necessary to consider memory, storage, processing, power, networking, and online media.
- Digital press specifications. Consider type resolution, online editing, online finishing, merge capabilities, storage, speed, page size and image size.
- Preflight results. Check most aspect of the preflight results.

CHAPTER 5
RESULTS

In this chapter it is introduced how the expert system works under some critical jobs. To do this a set of jobs from the Digital Publishing Facility at University of Puerto Rico were chosen, which let us test the system. The results of those decisions are compared with the decisions already made with the experimental jobs. To validate it was also created a set of random jobs, which are critical to the system, to determine the effectiveness of the decisions.

It is also important to notice that the setting of the system based on the characteristics of the Digital Publishing facility at UPRM. The framework of the system let each print shop to create its own rules about a job, characteristics of the output devices, and previous experiences on a certain job.

Our expert system is expected to infer if the document can be printed out of the input data received from the user or the metadata collected in any previous process of the Digital Publishing Workflow. This data comes from information such as the document type (brochure, thesis, poster,) page size, type of publication, and preflight.

This methodology will help for evaluation, validation and verification of the knowledge base of the DSS in DP. With this validation it can be verify the capability of producing empirically correct decisions. Also the precision of the system was

tested, allowing the system to replicate particular system parameters; the consistency of the advice and coverage of the knowledge base was also checked, for the decisional support system in digital publishing.

5.1 Scenario

5.1.1 Resources

In this section the test environment will be described. For the testing environment the Digital Publishing Facility at ECE was used. It is important for the system to know the characteristics of the resources since this is crucial to make the decision of the resources being used to determine the final result of a print job.

5.1.2 HPDesignjet 130

Color solutions for your workflow as :

- HP professional color technologies
- Automatic color calibration
- Automatic Pantone(tm) calibration (RIP)
- Offset emulation (RIP)
- CMYKplus (RIP)

Prints up to 24"/A1 size (HP Designjet 130/130nr) 4 different media paths for maximum media flexibility, including C-size tray and automatic roll-feed capability). The HP Designjet 30 printer series has 3 different media paths including a roll feed

. Support for RGB ICC profiles means one has the choice of managing color in the applications or via ICM profile or ColorSync tools. HP Software RIP Designed for creatives who need color management in a CMYK workflow, this optional RIP adds the capabilities designers need to generate accurate layouts, comps or concept proofs that integrate images with texts, logos or illustrations. A true Adobe

PostScript 3 solution, the RIP offers complete ICC profile management, including input and output profiles, as well as automatic PANTONE(r)2 calibration to easily and accurately print corporate or specific colors.

EFI Designer RIP for HP XL Offers in-RIP separations for the composite workflow, as well as the ability to define non-standard spot colors. It also offers job management tools including job ticket and control strip ability as well as job nesting for paper savings. One can easily emulate offset presses or other printing devices via ICC profiles, and CMYKplus support allows you to produce images perceptually consistent with offset output, but with the richer and more vibrant colors possible with a digital printer.

Third-party RIPs Capability to use other RIP Software.

5.1.3 HP Designjet 9500

Specification of the HP Designjet 9500 as shown in figure 5–2

5.1.4 HP Designjet 5500

5.1.5 Preflight Profile

For the testing environment it was decided to run one preflight profile for CMYK printers for professional printing. This PDF profile verifies whether a PDF file can be processed on a Digital Press black and white as a color printing. It also verifies the compliance with PDF X-1.

The preflight profile is summarized in the following list of characteristics. A complete list is available in the appendix.

- The PDF document requires at least Acrobar 5.0 (PDF 1.4).
- The PDF document can be encrypted.
- Check if page size or page orientation is different from page to page.
- Ignore area outside of TrimBox or BleedBox.
- Images Resolution of Color and gray scale images should not be is lower than 100 or higher than 450 pixels per inch.

Table 5–1: HP Designet 130nr

Printer Specification	
Print speed, color (normal mode)	B size: up to 4 min/page (Normal, glossy); D size: up to 11.9 min/page (Normal, glossy) Print speed, color (best quality mode) B size: up to 6 min/page (Best, glossy); D size: up to 17.5 min/page (Best, glossy) Print quality / technology.
Print technology	HP color thermal inkjet HP color layering technology with Optional software RIPs: Automatic PANTONE(tm) Calibration, PANTONE(tm) certified, ICC color profile support, Press emulations, Apple(r) ColorSync(r) Compatible, CMYKplus. HP Designjet 130gp only Eye One Display with monitor calibrator from GretagMacbeth.
Resolution	2400 dpi printing resolution, automatic closed-loop color calibration.
Ink technology	6-color, fade-resistant1, modular dye-ink supplies: C, M,Y, K, Lc, Lm.
Print speeds	A3/B-size - 6 mpp (in Best mode), A3/B-size - 4 mpp (in Normal mode), A1/D-size - 12 mpp in Normal mode and 17 mpp in Best mode, using an N5 file. Print speeds on HP Glossy media.
Paper handling	Up to 4 media feeding paths 70-sheet input tray; up to C+/A2+ Front manual single-sheet feed; up to 24.6 inches (625 mm) Single-sheet rear path for thick media; 0.02 in (0.4 mm) *Automatic roll feed with automatic cutter 2 50-sheet output tray Minimum paper size: 3 x 5.6 in (76 x 142 mm) Maximum paper size: 24.6 x 64 in (625 x 1625 mm) Automatic roll feed (width): up to 24 in (609.6 mm) 2 Maximum thickness (rear path): 0.02 in (0.4 mm).
Media sizes	Standard: Letter, legal, tabloid, executive, C, D, C+, D+, envelopes (A1, A1+, A2, A3, A4, B2, B3, B4, A2 metric oversize, A/B/C/D (English architectural), envelopes) Custom: Sheets - 3.0 x 5.6 to 24.6 x 64.0 in (76 x 142 to 625 x 1625 mm), automatic roll feed2 (width): 24 in (609.6 mm).
Media types	Paper (plain, coated, two-sided brochure paper, photo, glossy, semi-gloss photo, heavy weight), transparencies.
Printer languages	PCL3GUI RGB-24 bit Contone.
Memory	64 MB

Table 5–2: HP Designet 9500

Printer Specification	
Speed/monthly volume	Print speed, black (best quality mode) Up to 24 ppm Print speed, color (pages per minute) Up to 24 ppm.
Processor speed	500 MHz Recommended monthly volume, maximum 200,000 pages.
Print quality / technology	Resolution technology ImageREt 4800.
Paper handling / media	Paper trays, std. 9500hdn: 4 Paper trays, max. 9500hdn: 4 Input capacity, std. 9500hdn: Up to 3100 (20 lb bond) sheets Input capacity, max. 9500hdn: 3,100 sheets Standard envelope capacity Up to 10 envelopes envelopes Envelope feeder No.
Output capacity, std.	Up to 600 sheets Output capacity, max. 9500hdn: Up to 3500 sheets.
Duplex printing	Available 9500hdn: Automatic (standard).
Media sizes, std.	Letter, letter-R, legal, foolscap (8.5 x 13 in), executive, tabloid (11 x 17 in.), envelopes (No. 10, C5, B5, DL, Monarch).
Media sizes, custom	9500hdn: Tray 1: 3.9 x 7.5 to 12.1 x 18.5 in; Tray 2, 3: 5.8 x 8.3 to 11.7 x 17 in; Tray 4: 7.2 x 8.3 to 11.7 x 17 in.
Memory / print languages	Memory, std. 9500hdn: 288 MB Memory, max. 416 MB.
Hard disk	9500hdn: Standard, 20 GB.
Print languages	std. PostScript(r) 3(tm) emulation, PCL 5c, PCL 6, direct PDF Typefaces 80 TrueType(tm) internal scalable in PCL, 80 TrueType internal scalable in PS; Euro symbol supported.

Table 5–3: HP Designet 5500

Printer Specification	
Media sizes, std.	8.3- to 60-in wide sheets; 24-, 36-, 42-, 60-in rolls.
Maximum print length	300 ft.
Ink types	Dye-based.
Print technology	HP Thermal Inkjet.
Print quality, black	1200 x 600 dpi (on glossy media).
Print quality, color	1200 x 600 dpi (on glossy media).
Print languages, std.	HP-GL/2, HP RTL, TIFF 6.0, JPEG, CALS/G4 Memory, std.
128 MB, 40 GB hard disk	
Print speed, fast mode	569 sq ft/hr (maximum speed).
Print speed, normal mode	189 sq ft/hr (production mode); 100 sq ft/hr (productivity mode).
Color print speed, best	76 sq ft/hr (maximum quality).
Resolution	1200 x 600 dpi printing resolution Maximum resolution (with enhanced IQ on): 1200 x 600 dpi (on glossy media).

- Resolution of Bitmap images should not be lower than 550 or higher than 3600.
- Check if images use lossy compression.
- Check if the objects on the page use RGB or CMYK.
- Check if fonts are not embedded.
- Check if the fonts are embedded as a subset or completely.
- Check if font type is Type 1 , TrueType , Type 3.
- Check if PDFC documents are compliant with PDF/X-1.

5.1.6 Scenario 1

This job is divided in two formats: one full color color 8.5 x 11 in using 450dpi and the poster same in 11x17 in using 450 dpi shown in figure 5–1. The client wants a high quality job in the 11 x 17 photography glossy paper , and a medium quality in 8.5x11. We started with the 8.5x11 running preflight we notice that it was a resample from the 11x17. Preflight results are shown in figure 5–5 .As shown in figure 5–3 , the metadata of the PDF file , and the intent, is described. In this

Table 5–4: PDF Preflight Result Scenario 1

Serverity	*Description*
Error	Found objects with transparency settings
Error	ICCBased is used
Caution	Compression ratio of image is more than 10.0
Fixed	Changed image resolution (1x)
Fixed	Changed page box layout in conformity with the press layout specifications

figure we show how the system as the question about the job, each question was fired, depends in the previous answer.

For the final decision the rule engine decides what rule to fire, , the rules add, remove , and modify facts in the working memory to make a final decision as shown in Figure 5–2. To make this decisions it took the expert system 4 rules fired, and to make the recommendation 6 more. Once it runs the DSS, the system accepted the job in both posters jobs, since it was a professional PDF file, and we can print it in any printer already described.

Figure 5–1: PDF Scenario 1

5.1.7 Scenario 3

We send the metadata is sent and already captured to the DSS: interestingly enough the job was a poster 56x28" shown in figure 5–5, which has of an unusual size. We run the preflight process. The results are shown in table 5–6 and the XMP metadata in figure 5–4. Here, the PDF metadata module was tested, and from the preflight result the system sent good results to print this poster in a wide format. Once the PDF metadata module was activated, the DSS results in failure

```
FIRE 38 recommend::CMYKprocess Accepted f-37, f-33, f-27
 ==> f-58 (MAIN::recommendation (printjob CMYKprocess) (explanation "Based on the Generic Press profile, this PDF
Profile assumes the intended output device uses pure process color."))
FIRE 39 recommend::fontType f-33
 ==> f-59 (MAIN::recommendation (printjob GenericOffice) (explanation "Font Type Change : Check the PDF file for the
presence
 of TrueType  TrueType fonts are scalable fonts that are built into Windows and Mac OS,
 and print well on both PostScript and non-PostScript printers.
 They are used widely, and are integrated in almost
  all desktop office software applications for the Windows and Mac operating systems. However, some professional
prepress service providers


"))
FIRE 40 recommend::paperSizePoster f-25
 ==> f-60 (MAIN::recommendation (printjob Generic) (explanation "its poster 16x20 "))
FIRE 41 recommend::typeofjob f-23
 ==> f-61 (MAIN::recommendation (printjob Generic) (explanation "All poster must use Generic Profile"))
FIRE 42 recommend::combine-recommendations f-61, f-60
 <== f-60 (MAIN::recommendation (printjob Generic) (explanation "its poster 16x20 "))
 <== f-61 (MAIN::recommendation (printjob Generic) (explanation "All poster must use Generic Profile
its poster 16x20 "))
FIRE 43 report::sort-and-print f-58,

JOB Decision:           CMYKprocess
FIRE 44 report::request-profileExplanation f-0
 ==> f-62 (MAIN::ask profileExplanation)
FIRE 45 ask::ask-question-by-id f-13,, f-62
Do you want to see the reason for choosing the profile? (yes or no) yes
 ==> f-63 (MAIN::answer (ident profileExplanation) (text yes))
 <== f-62 (MAIN::ask profileExplanation)
FIRE 46 report::Reason f-63, f-58,

Reason:                Based on the Generic Press profile, this PDF Profile assumes the intended output device uses
pure process color.

 <== f-58 (MAIN::recommendation (printjob CMYKprocess) (explanation "Based on the Generic Press profile, this PDF
Profile assumes the intended output device uses pure process color."))
FIRE 47 report::Reason f-63, f-61,

Reason:                All poster must use Generic Profile
its poster 16x20

 <== f-61 (MAIN::recommendation (printjob Generic) (explanation "All poster must use Generic Profile
its poster 16x20 "))
FIRE 48 report::Reason f-63, f-59,

Reason:                Font Type Change : Check the PDF file for the presence
 of TrueType  TrueType fonts are scalable fonts that are built into Windows and Mac OS,
 and print well on both PostScript and non-PostScript printers.
 They are used widely, and are integrated in almost
  all desktop office software applications for the Windows and Mac operating systems. However, some professional
prepress service providers

 <== f-59 (MAIN::recommendation (printjob GenericOffice) (explanation "Font Type Change : Check the PDF file for the
presence
 of TrueType  TrueType fonts are scalable fonts that are built into Windows and Mac OS,
 and print well on both PostScript and non-PostScript printers.
 They are used widely, and are integrated in almost
  all desktop office software applications for the Windows and Mac operating systems. However, some professional
prepress service providers
```

Figure 5–2: Scenario 2 DSS Accepted Job

```
FIRE 2 interview_intent::request-typeofjob f-0
 ==> f-22 (MAIN::ask typeofjob)
FIRE 3 ask::ask-question-by-id f-1,, f-22
What kind of job is , Poster,Brochure,Magazine>  Poster
 ==> f-23 (MAIN::answer (ident typeofjob) (text Poster))
 <== f-22 (MAIN::ask typeofjob)
FIRE 4 interview_intent::request-paperSizePoster f-23
 ==> f-24 (MAIN::ask paperSizePoster)
FIRE 5 ask::ask-question-by-id f-7,, f-24
Poster Size is 11x17 , 18x24 or 20x30:  11x17
 ==> f-25 (MAIN::answer (ident paperSizePoster) (text 11x17))
 <== f-24 (MAIN::ask paperSizePoster)
FIRE 6 interview_intent::request-colorinfo f-0
 ==> f-26 (MAIN::ask colorinfo)
FIRE 7 ask::ask-question-by-id f-6,, f-26
Intent is a Color or BW ? Color
 ==> f-27 (MAIN::answer (ident colorinfo) (text Color))
 <== f-26 (MAIN::ask colorinfo)
FIRE 8 interview_intent::request-graphicContent f-0
 ==> f-28 (MAIN::ask graphicContent)
FIRE 9 ask::ask-question-by-id f-9,, f-28
Does the document has any type of Graphics?  (yes or no) yes
 ==> f-29 (MAIN::answer (ident graphicContent) (text yes))
 <== f-28 (MAIN::ask graphicContent)
```

```
FIRE 19 ask::ask-question-by-id f-8,, f-38
Printer type is Professional or Home/Office:  Professional
 ==> f-39 (MAIN::answer (ident printerType) (text Professional))
 <== f-38 (MAIN::ask printerType)
FIRE 20 interview_intent::request-spotcolor f-39
 ==> f-40 (MAIN::ask spotcolor)
FIRE 21 ask::ask-question-by-id f-11,, f-40
Does the document contains spot color?   (yes or no) no
 ==> f-41 (MAIN::answer (ident spotcolor) (text no))
 <== f-40 (MAIN::ask spotcolor)
FIRE 22 metadata_pdf::request-pdf_Title f-0
 ==> f-42 (MAIN::ask pdf_Title)
FIRE 23 ask::ask-question-by-id f-14,, f-42
PDF::Title Spice IEEE Poster
 ==> f-43 (MAIN::answer (ident pdf_Title) (text Spice))
 <== f-42 (MAIN::ask pdf_Title)
FIRE 24 metadata_pdf::request-pdf_Trapped f-0
 ==> f-44 (MAIN::ask pdf_Trapped)
FIRE 25 ask::ask-question-by-id f-21,, f-44
PDF::Trapped yes
```

Figure 5–3: Expert System Scenario 2 Metadata Capture

Table 5–5: PDF Preflight Result Scenario 2

Serverity	*Description*
Error	Found objects with transparency settings (14x).
Fixed	Changed flatness (188x).
Error	ICCBased is used (10442x).
Error	Effective resolution of color or grayscale image is less than 280 dpi.
Fixed	Changed page box layout in conformity with the press layout specifications.
Caution	Text point size 2.2 is less than 4.0 (72x).
Caution	X and Y scaling of image differs 1.

Table 5–6: PDF Preflight Result Scenario 2

Serverity	*Description*
Fixed	Changed flatness (1x).
Error	ICCBased is used.
Error	Effective resolution of color or grayscale image is less than 280 dpi.
Fixed	Changed page box layout in conformity with the press layout specifications.

to send this job through the rest of the workflow because the job was sent as an 8x11 in. The DSS advice was not to send this job based on the characteristics of the intended output. The recommendation of the system was to send back the PDF and create it in an better quality and the right size for a better resolution and a better quality of the job.

5.1.8 Scenario 2

This job was a 11 x17 poster shown in figure 5–6, where the quality of the job was low.

Based on the previous information, and the low quality expected by the client, the Print job can be done in any of the output devices as mentioned before in section 5.1.1.

It was tested this job but changing the quality expected of the print job. To test our system we changed the Client profile was changed to a Medium Quality, and the Decision Support System sent a failure notification, since it has a 280 DPI

```
  <rdf:Description rdf:about=''
 xmlns:pdf='http://ns.adobe.com/pdf/1.3/'>
 <pdf:Producer>Acrobat Distiller 7.0.5 (Windows)</pdf:Producer>
</rdf:Description>

  <rdf:Description rdf:about=''
 xmlns:xap='http://ns.adobe.com/xap/1.0/'>
 <xap:CreatorTool>Acrobat PDFMaker 7.0.5 for PowerPoint</xap:CreatorTool>
 <xap:ModifyDate>2006-02-23T14:56:10-08:00</xap:ModifyDate>
 <xap:CreateDate>2006-02-23T14:55:46-08:00</xap:CreateDate>
 <xap:MetadataDate>2006-02-23T14:56:10-08:00</xap:MetadataDate>
</rdf:Description>

   <rdf:Description rdf:about=''
 xmlns:dc='http://purl.org/dc/elements/1.1/'>
 <dc:format>application/pdf</dc:format>
 <dc:title>
  <rdf:Alt>
   <rdf:li xml:lang='x-default'>Slide 1</rdf:li>
  </rdf:Alt>
 </dc:title>
 <dc:creator>
  <rdf:Seq>
   <rdf:li>Sunke</rdf:li>
  </rdf:Seq>
</dc:creato
   <rdf:Szie>
   <rdf:li>8x11</rdf:li>
  </rdf:Size>

</rdf:Description>
```

Figure 5–4: XMP Metadata for Scenario 1

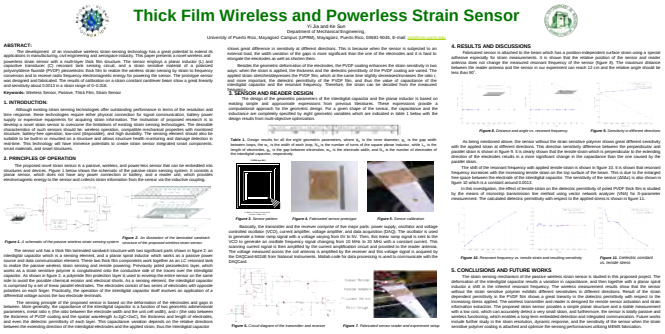

Figure 5–5: Scenario 2 Print Job Poster

Figure 5–6: Scenario 2 Print job

image in a 11x17 job. This error is easy to detect due to the fact that the image takes a large area of the composition of the job as shown in figure 5–6.

5.2 Conclusion

It is proved that under these circumstances the knowledge base was constructed properly, which led to validate the KB. The results obtained by the DSS were validated in the Digital Publishing facility at UPRM, where the DSS can performed the same functions as an expert, but it does not eliminates any process, since the Digital Publishing print shops are different from each other, an expert is needed to add more knowledge to the system to achieve a better performance and more detailed in the final decisions.

CHAPTER 6
CONCLUSION

The knowledge base system expresses the knowledge within an organization, and the use of ruled base system adds the capability to manage this knowledge base system and provides a conclusion that helps the workflow engine or the operator to make formal decisions on a certain process. Through the use of knowledge base systems, more capability can be added in the decision process of digital publishing workflow system. This methodology can be applied in other steps of the workflow system to discover new information that can be acquired with the knowledge base system. This information in the ontology and the knowledge base systems can represent the domain knowledge that can be re-used, and it can also analyze domain knowledge to be implemented in other applications.

that directs and connects each step in the workflow help to eliminate time- consuming manual and redundant tasks, reduces errors, resulting in more predictable, higher quality output. Increases productivity and speeds job turnaround enabling printers to increase capacity to determine the results of a certain job. Adds value by integrating print production with enterprise business operation.

6.1 Future Work

For future investigation the following research topics were proposed:

- To include metadata of other processes involved in digital publishing. This includes the expansion of the ontology and the knowledge representation.
- To implement a Graphic User interface to modify the knowledge base system.

- To Develop the decisional support system as a Web Service.
- To Integrate with a workflow engine.
- To Explore Enterprise applications.
- To Explore Rule formats as RuleML. .
- To Deploy Expert System as Servlet.
- To include metadata of other processes involved in digital publishing. This includes the expansion of the ontology and the knowledge representation.
- To implement and Graphic User interface that interacts with the rules in the knowledge base system that serves the expert to add more knowledge.
- To Develop a Web Service of decisional support system.
- Integrate with a workflow engine.

APPENDICES

APPENDIX A
PREFLIGHT PROCESS AND METADATA ARCHIVES

REFERENCE LIST

[1] H. Johnson. *Mastering Digital Printing.* Thomson, second edition edition, 2005.

[2] N. Santiago F.Vega T.Avellanet G.Chaparro W.Lozano A.Pereira H.Santos-Villalobos W.Rivera, M.Rodriguez-Martinez. Towards development of concepts and algorithms to enable automated digital publishing workflows",. 2005.

[3] T. Finin T. Gruber R. Patil T. Senator R.Nehces, R . Fikes and W.R. Swartout. Enabling technology for knowledge sharing. *AI magazine, 12(3), 36-56*, 1991.

[4] Gruber T.R. A translation approach to protable ontology specifications. *Knowledge Acquistions 5(2), 199-220*, 1993.

[5] P. Borst. Engineering ontologies. *International Journal of Human-Computer Studies 46(2-3), 265-406*, 1997.

[6] K. Knoght B. Swartout, R. Patil and T.Russ. Towards distributed use of large-scale ontologies. *In Proceedings of 10th Knowledge Acquisition Workshop pp. 32.1-32.19*, 1996.

[7] A. Farquhar R. Fikes and J. Rice. The ontolingua server: a tool for collaborative ontology construction. 46, 707-728. *Journal of Human-Computer Studies*, 1997.

[8] N. Singh. Unifying heterogeneous information models. *Commun. ACM 41,5 p. 37-44*, 1988.

[9] P. Clark M. Uschold, R. Jasper. Three approaches for knowledge sharing. *KAW*, 1999.

[10] King Uschold. Towards a methodology for building ontologies. *IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing.. Montreal,Canada,pp 6.1-6.10*, 1995.

[11] E. Franconi. A semantic approach for schema evolution and versioning in object oriented dataqbases. *Computational Logic 2000 pp 1048-1062*, 2000.

[12] A.J. Gonzalez and D.D. Dankel. The engineering of knowledge-base systems. *Prentice-Hall*, 1993.

[13] E.M Award. Building knowledge automation expert systems with exsys. *Corvid, Exsys Inc USA*, 2003.

[14] Michael L. Kepler. *The Handbook of Digital Publishing*. Vol II Prentice-Hall,, second edition edition, 2001.

[15] Jdf specification version 1.3.

[16] Print On Demand Initiative (PODi). Best practices in digital print. *Fifth Edition*, 2005.

[17] *Adobe Systems Incorporated PDF Reference (Second Edition) Verssion 1.3*. Addison-Wesley, July 2000.

[18]

[19] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers*, 2004.

[20] L.Leff. Automated reasoning with legal xml documents. In *ICAIL '01: Proceedings of the 8th international conference on Artificial intelligence and law*, pages 215–216, New York, NY, USA, 2001. ACM Press.

[21] J. Euzenat M. Hori and P.F. Patel. Owl web ontology language xml presentation syntax. *W3C Note, Available http://www.w3.org/TR/owl-xmlsyntax*, 11 june 2003.

[22] L.Ma, Z.Su, Y.Pan, L.Zhang, and T.Liu. Rstar: an rdf storage and query system for enterprise resource management. In *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management*, pages 484–491, New York, NY, USA, 2004. ACM Press.

[23] N.Guarino and P.Giaretta. Ontologies and knowledge bases: Towards a terminological clarifications. *In Towarrd Very large Knowledge Bases: Knowledge Building and Knowledge Sharing 25-32 ISO Press: Amsterdam, The Netherlands*, 1995.

[24] N.Noy and C.Hafner. The state of the art in ontology design: A survey and comparative review. *AI Magazine, 18 ,3 , 53-74*, 1997.

[25] V.Ver and N.Mars. Bottom-up construction of ontologies. ieee. *Transactions on Knowledge and Data Engineering, 10, 4, 513-526*, 1998.

[26] D.Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications to the ACM, 38,11, 33-38*, 1995.

[27] G. Zuniga. Ontology: its transformation from philosophy to information systems. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 187–197, New York, NY, USA, 2001. ACM Press.

[28] M.Fernandez, N.Onose, and J.Simon. Yoo-hoo!: building a presence service with xquery and wsdl. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 911–912, New York, NY, USA, 2004. ACM Press.

[29] W3C. Web services description languages (wsdl).

[30] W. Nejdl, B. Wolf, C. Qu, S. Decker, M.Sintek, A.Naeve, M.Nilsson, M.Palm, and T.Risch. Edutella: a p2p networking infrastructure based on rdf. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 604–615, New York, NY, USA, 2002. ACM Press.