A Regulation-Centric, Logic-Based Compliance Assistance Framework

Shawn L. Kerrigan and Kincho H. Law
Department of Civil and Environmental Engineering
Stanford University
Stanford, CA 94305-4020
Email Contact: law@stanford.edu

Abstract

This paper describes the development of a logic based regulation compliance assistance system that builds upon an XML (eXtendable Markup Language) framework. First, a document repository containing federal regulations and supplemental documents, and an XML framework for representing regulations and associated metadata are briefly discussed. The prototype effort for the regulation assistance system focuses on federal environmental regulations and related documents. The compliance assistance system is illustrated in the domain of used oil management. The overall objective is to develop a formal infrastructure for regulatory information management and compliance assistance.

Keywords: XML (eXtendible Markup Language), compliance assistance, regulation management, document repository, environmental regulations

1 Introduction

In the United States, both federal and state, as well as local governments, have strict regulations imposed on the protection of the environment. Environmental regulations are complex and voluminous, which can be disproportionately burdensome on small businesses. A significant amount of regulatory information is available online through various regulatory portals, and the coverage of online material continues to grow. However, most of the current online portals are primarily designed for displaying the information for experienced users and are difficult to use for further processing. Information technology (IT), if properly designed and developed, has the potential to help the access and retrieval of relevant information and to facilitate the compliance process. The REGNET research project at Stanford University aims to develop a formal infrastructure for regulatory information management and compliance assistance.

There has been a push in the United States by the executive office for government agencies to put more emphasis on compliance assistance in lieu of enforcement to encourage companies to comply with regulations (Van Wert 2002, National Compliance Assistance Providers Forum 2002). Towards this end, specialized programs using expert system technologies have been built to assist users in understanding regulation requirements for particular circumstances (Botkin 2002). One significant limitation of the systems currently available online is that they do not directly map to the regulations or legal documents that they represent. The failure to map to the source documents creates four significant disadvantages. First, because users do not see the regulation text as they interact with the system, users may have difficulty understanding the results produced by the system. Second, since users do not see the regulations during processing they may have trouble learning how the regulation works, and may have difficulty re-tracing the results of the system on paper for validation purposes. Third, since users cannot track how the system is proceeding with its analysis, they will have trouble investigating background information on issues or questions the system raises. Fourth, updating the system as the regulation changes is difficult, since without a mapping between the regulation and the rules in the system it may not be clear what parts of the system need to be changed when the regulation is altered.

This paper describes our research on developing a compliance assistance infrastructure that builds upon an XML (eXtendible Markup Language) regulation framework. By using a regulation-centric approach to structuring a compliance assistance system around the regulation itself, this infrastructure allows clear linkages to the regulation text, thus overcoming many of the limitations of the systems currently in use. In particular, because all encoded regulation rules are tied to particular regulation provisions, it is straightforward to map the compliance process to the provisions.

We first briefly describe a document repository containing federal environmental regulations and supplemental documents, and an XML framework for representing regulations and associated metadata. We then describe in detail the prototype effort for the regulation assistance system, along with a discussion of how the regulation assistance system may fit in the broader compliance process, for example, linking with online guidance systems. The regulation assistance system is illustrated in the domain of used oil management.

2 Document Repository and XML Regulation Framework

One objective of the REGNET information infrastructure is to develop a document repository for environmental regulations. The scope of our current prototype development covers Title 40 of the US Code of Federal Regulations (40 CFR): Protection of the Environment, along with selected supplementary and supportive documents that focus on regulations covering hazardous waste and the management of used oil. Supplemental documents are important because they often contain information that is necessary for the accurate interpretation of the federal regulation(s) to which they refer (Heffron and McFeeley 1983). Supplementary documents may come in the form of administrative decisions, guidance documents, court cases, letters from the general counsel and letters of interpretation from the Environmental Protection Agency (EPA). The REGNET document repository is designed to make these important documents more accessible. The contents of the repository are available through the mediation of one or more searchable concept hierarchies, or through a regulation assistance system (Kerrigan 2003).

We have developed an XML framework for environmental regulations. XML (eXtendible Markup Language) is a meta-markup language that consists of a set of rules for creating semantic tags used to describe data elements and provides a mechanism to describe a hierarchy of elements that forms an object structure. The XML framework is regulation centric and includes XML tags for each level of regulation text – for example part, subpart, section or subsection – that mirrors the standard structure of regulations. This framework results in a hierarchical structure for the regulations, with regulation text attached throughout. Figure 1 shows how a regulation can be decomposed into a hierarchical tree structure. Figure 2 shows an abbreviated sample of how we represent this hierarchical structure in XML. Parsing systems have been built to transform the federal regulations from Portable Document Format (PDF) and HTML into REGNET's XML framework (Kerrigan 2003). These parsers use pattern-matching approaches to identify the structure of a regulation and create an explicit XML structure around the regulation text. With XML, it is possible to augment a regulation with various types of annotation and regulation-specific metadata rather than to simply structure the regulation according to how it should be displayed. With respect to the document repository, the metadata types currently added to the regulation framework include concept tags, reference tags and definition tags.

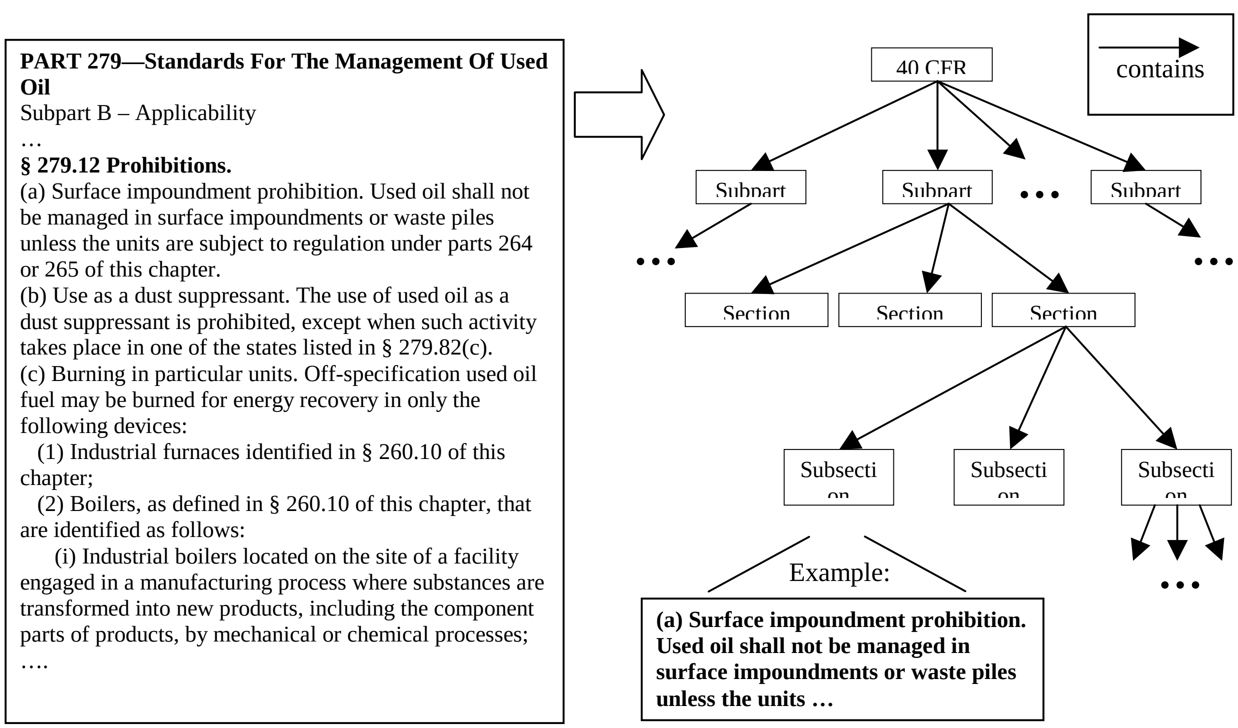


Figure 1. Decomposition of regulation into a hierarchical tree structure

```
<regulation id="40.cfr.279" name="Standards For The Management Of Used Oil" type="US Federal">
  …
  <regElement id="40.cfr.279.B" name="Subpart B">
    …
    < regElement id="40.cfr.279.12" name="Prohibitions">
      < regElement id="40.cfr.279.12.a" name="Surface Impoundment prohibition">
        <regText>
          <paragraph>
          Used oil shall not be managed in surface impoundments or waste piles…
          </paragraph>
        </regText>
      </regElement>
      <regElement id="40.cfr.279.12.b" name="Use as a dust suppressant">
      …
      </regElement>
    …
    </regElement>
    …
  </regElement>
  …
</regulation>
```

Figure 2. Abbreviated XML representation of regulation tree structure

The concept tags allow dynamically generating links to related supporting documents in the document repository. This is useful because supporting documents and regulations may not directly reference each other even when they address the same topic. The automatic application of concept tags to the XML framework means that as new supporting documents are added to the document repository, regulations stored in the framework can automatically be linked to them via the terms that they share in common. Concept tags can be generated “semi-automatically” using existing text mining and information retrieval tools (Kerrigan 2003). Currently, we use software from Semio Corp. to help extract, clean and define over 65,000 concepts for the 40 CFR regulations and to categorize the concepts according to different interests and applications.

Regulation provisions tend to contain a large number of casual English references to other provisions. These references are cumbersome to look up manually, and they reduce the readability of the regulation text itself. Simple references (for example, “as stated in 40 CFR section 262.14(a)(2)”) and complex references (for example, “the requirements in subparts G through I of this part”) exist throughout regulations. Given the large volume of regulations, a manual translation of references would be too time consuming. A parsing system has been developed using a context-free grammar and a semantic representation/interpretation system that is capable of tagging regulation provisions with the list of references they contain (Kerrigan 2003). Instead of building hyperlinks, which tie the reference to a particular source for the referred document, the reference tags provide a complete specification for what regulation provision is referenced. Where the regulation is located is not specified so that a viewing system may select from any document repository of regulations to retrieve the referenced provision. This provides more flexibility than a rigid hyperlink structure for maintenance and scalability.

The large number of domain-specific terms and acronyms that appear in regulations can make regulation text difficult for novices to understand. We standardize all definitions with XML elements, which allow regulation-viewing systems to incorporate explicit definitions of terms and acronyms into their user interfaces.

3 Regulation Assistance System

This section discusses the development of a regulation assistance system (RAS), which is the focus of this paper. First, predicate logic is briefly introduced as a form of metadata. Second, additional metadata added to the XML regulations described earlier to enable a logic-based compliance assistance system are discussed. Third, the algorithms used for compliance checking are presented.

3.1 Logic-Based Metadata for Compliance Assistance

This section introduces the types of metadata specifically implemented for the web-based compliance assistance system. Besides the concept, reference and definition tags, we add logic and control processing metadata to the XML regulation framework. The logic metadata represents a rule or concept from a regulation using First Order Predicate Calculus (FOPC) logic sentences. The user interface with compliance questions and possible answers is also encoded in FOPC logic sentences as metadata in the XML structure. Control processing metadata provides information about which provisions of a regulation need to be checked for compliance. For the purpose of demonstration, a federal used oil regulation (40 CFR 279) has been manually tagged with regulation logic metadata, with user-interface logic metadata, and with control processing metadata.

3.1.1 Predicate Logic

Symbolic logic is a representational formalism used to describe concepts, ideas and knowledge. The formal representation of knowledge can be used to reason about the information and to draw new conclusions or look for contradictions. Use of formal symbolic logic can also be used to communicate information between systems (Genesereth 1992). First Order Predicate Calculus (FOPC) is a symbolic logic language that will be briefly introduced in this section. For a more in-depth treatment of this subject please refer to (Zohar and Waldinger 1993).

Predicate logic is similar to propositional logic, but allows quantification and the usage of objects. Predicate logic sentences are composed of connectives, truth symbols (*true* or *false*), constants, variables, predicate symbols and function symbols. Constants and variables denote objects. Predicates define relationships between objects. Functions define functions on the objects. Predicates and functions have defined arities that are the number of arguments or terms associated with their use. Terms may be constants, variables or function expressions. The connectives between elements in a predicate logic sentence can be “and” ($\wedge$), “or” ($\vee$), “not” ($\neg$), “implies” ($\Rightarrow$), or “equivalent” ($\equiv$). Quantifiers are used to quantify predicate logic variables as universally or existentially quantified. There exist rules that may be used to perform proofs using these elements of predicate logic.

We use FOPC to model regulations in this research work because it offers a flexible, standardized, and computable representation. The choice of FOPC also introduces a great deal of flexibility for the choice of a reasoning system, since there are many reasoners available for working with FOPC. The current system, using FOPC, cannot precisely model the regulations. FOPC does, however, allow us to model the regulation rules in a simplified form that is sufficient for constructing a system to guide users through regulations and identify potential conflicts with the regulation rules.

In order to represent logic statements in an XML-based representation, there are syntactic limitations that must be met to comply with the XML standard. For example, XML elements are defined by the XML standard to start with “<”, as in “<regText>”. This conflicts with the standard logic syntax used for reverse implications, “<-“, and equivalences, “<->”. A simple substitution of text provides the solution for this problem, where the illegal XML character sequences are replaced with legal ones.

The substitutions currently being used to represent FOPC in an XML compliant syntax are shown in Table 1. Note that the substitutions for “->” and “|” are not necessitated by XML standards, but are done so that the XML logic uses a consistent representation formalism. The substitutions for “<-”, “<->”, and “&” are required by XML standards. The substitutions are reversed by the logic processing systems that read the XML regulation so that the standard syntax is used when providing the data to a logic reasoner. The XML compliant substitutions also become reserved words in the logic representation language. Since the words in the right column of Table 1 will be substituted with the logic symbols in the left column, words in the right-hand side of the table are reserved words that cannot be used for logic predicates or function names.

Table 1. Substitutions for XML compliant logic sentences

Standard logic syntax	**XML compliant substitution**
`->`	ForwardImplies
`<-`	ReverseImplies
`<->`	EquivalentTo
&	AND
|	OR

3.1.2 Basic Logic Elements

Logic can be added to the XML-based regulation document to facilitate manipulation and interpretation of the document. Internal contradictions within the regulation can be checked for, contradictions between regulation documents can be identified, and compliance checking systems can be built to verify that a user is in compliance with the regulation. The approach of tagging XML structured regulations with FOPC introduces an open platform consisting of structured text and embedded logic. Logic elements can be added to the XML structure within the regElement XML elements. The logic elements are denoted by “logic” tags, and may contain either logicSentence or logicOption elements.

The logicSentence elements are used to tag regulation provisions throughout the document to represent their logical meaning. For example, tagging the root regulation element with a logicSentence element specifies that the logic sentence should be applied to the entire document. The logicSentence elements are generally used to define the rules and concepts expressed in a regulation. Figure 3 illustrates a logicSentence element, where the logicSentence element describes a rule that used oil is not a valid dust suppressant. The rule states that for all objects “o”, if “o” is used oil then “o” is not a valid dust suppressant. The use of “ForwardImplies” instead of the more commonly used logic syntax “`->`” is necessitated by the XML standard.

The logicOption element is used to build a structured question and answer system that constructs logic sentences based on the user’s input. A logicOption element contains one question element to prompt a user for input, and one or more answer elements to represent the possible answers and the associated logic statements for the question. The user interacts with the system in plain English, but the answers are mapped to logic statements so that they can be used for compliance checking. Figure 4 illustrates the usage of a logicOption element that assists with gathering user input. This particular element maps the user’s response to a question about the use of used oil to logic statements that reflect the user’s answer.

```
<logicSentence>
  all _o  (usedOil(_o) ForwardImplies -(dustSuppressant(_o))).
</logicSentence>
```

Figure 3. Illustration of the logicSentence element

```
<logicOption>
  <question>
    Is the used oil used as a dust suppressant?
  </question>
  <logicOpt answer = "yes">
    <logic_ans>
      (usedOil(oil1) AND dustSuppressant(oil1)).
    </logic_ans>
  </logicOpt>
  <logicOpt answer = "no">
    <logic_ans>
      (usedOil(oil1) AND (-(dustSuppressant(oil1)))).
    </logic_ans>
  </logicOpt>
</logicOption>
```

Figure 4. Illustration of a logicOption element

3.1.3 Simple Control Processing Elements

There are three basic control elements, namely: goto, switchTo, or end. These three control elements allow regulation designers to specify what regulation provisions may or may not need to be investigated. While not FOPC in nature, control elements provide processing logic and therefore may be used within the logic XML elements.

The goto control element specifies a regulation provision that the system should process next; returning to the current provision once the specified provision has completed its check. The goto element is useful when it is necessary to check additional regulation provisions without abandoning the current line of processing. For example, frequently a regulation provision will refer readers to another regulation provision that should be read before continuing.

Similarly, the switchTo element specifies a regulation provision to process next, but processing should not return to the current provision once the specified provision has completed its check. This is useful when a regulation provision specifies some conditions under which a different regulation provision will apply.

Figure 5 illustrates the goto and switchTo elements. This example instructs the system to process Section 279.65, and once processing for that section is complete to switch processing to Section 279.73. The control attribute “target” is used to direct processing control to move to a reference. For example, target = “40.cfr.279.65”, refers the compliance processing system to Section 279.65 in 40 CFR.

Figure 6 illustrates an end element, which signals that the specified provision should not be investigated further. Since regulation checks may be done at any level of the regulation document, it is important to specify a target reference for the end element. For example, if a compliance check is initiated at Section 40 CFR 279.12 and an end element is encountered that specifies that Subsection 40 CFR 279.12(a) is complete, as shown in Figure 6(a), the check against the higher level Section 40 CFR 279.12 is not finished and should continue with the next subsection it contains. On the other hand, if a compliance check is initiated at Subsection 40 CFR 279.12(a) and an end element is encountered that specifies that the higher level Section 40 CFR 279.12 is complete, as shown in Figure 6(b), processing of Subsection 40 CFR 279.12(a) should stop because the subsection is contained within Section 279.12.

3.1.4 Conditional Control Processing Elements

The simple control elements specify immediate and unconditional changes in the processing control. The goto, switchTo and end elements all define actions that should be executed immediately, without regard for any information contained in the logic sentences already gathered by the system. What we also need is a way to specify that under certain logical conditions a regulation provision should be checked or the process should not proceed any further. The logic constructions used are of the form “X implies provision Y applies” and “X implies provision Y does not apply”. Figure 7 shows a regulation provision where the applicability of a regulation subpart depends partly on information that may not be currently available. In this example the section 40 CFR 279.23 may not have been encountered yet, so it may not be possible to determine if subpart G should be processed.

Conditional control statements are written in standard logic sentences, with a forward implication which contains the predicate “provApplies” or “provDoesNotApply” as the consequent of the implication. The two predicates indicate whether the provision contained by the predicate should or should not be processed. Figure 8 illustrates how the provision in Figure 7 can be represented in FOPC. Using this representation, if a company is a used oil generator that also burns the used oil it generates, Subpart G

```
<control>
  <goto target = "40.cfr.279.65" />
  <switchTo target = "40.cfr.279.73" />
</control>
```

Figure 5. Illustration of the goto and switchTo elements

```
<control>
  <end target = "40.cfr.279.12.a" />
</control>
```

6(a)

```
<control>
  <end target = "40.cfr.279.12" />
</control>
```

6(b)

Figure 6. Illustration of the end element

40.cfr.279.20.b.3 states:
Generators who burn off-specification used oil for energy recovery, except under the on-site space heater provisions of §279.23, must also comply with subpart G of this part.

Figure 7. A provision from 40 CFR 279

```
<logicSentence>
   all _client _oil ((generator(_client) AND usedOil(_oil) AND
   burnsForEnergy(_client, _oil) AND -satisfied(40_cfr_279_23))
   ForwardImplies provApplies(40_cfr_279_G)).
</logicSentence>
```

Figure 8. Logic representation for conditional control statement

will apply if 40 CFR 279.23 is not satisfied. The complete logic representation for 40 CFR 279.20(b)(3) should include an element that directs the system to check for compliance with 40 CFR 279.23 if someone is a used oil generator who also burns used oil.

3.2 Logic-Based Compliance System

This section describes in detail how the XML regulation framework can be used to support compliance assistance services.

Figure 9 shows the organization of the RAS system which includes a web interface (RASweb), the RCCsession component, and an automated-deduction program. The system is implemented with a web interface built upon a compliance-checking component (RCCsession). The RCCsession component controls the process used to check for violations. The process begins by parsing the XML-structured regulation to extract the information necessary to run a compliance check against the document. This information includes the logic metadata as well as the control processing metadata. The RCCsession follows the control processing metadata in the XML-regulation structure and manages lists of regulation rules and the associated user responses. A user response is mapped to corresponding FOPC user-interface logic sentences for that response according to the associated logicOption element's contents. Whenever the RCCsession component has a set of logic sentences that it needs to check for contradictions, an input file containing those logic sentences is generated and passed onto an automated deduction system to check for a proof. The RCCsession then reads the results produced by the automated deduction system to see if a proof has been found and then takes whatever actions necessary to continue the compliance checking procedure. In short, the RCCsession controls the flow of processing while the theorem prover is used to check for logical contradictions in the background. The system design is such that any FOPC theorem prover can be used to perform the logical contradiction checks. Presently, we employ Otter, a publicly available theorem prover developed at Argonne National Laboratory (McCune 1994).

Figure 10 shows the basic compliance assistance process. First, the XML regulation is verified. Second, logic sentences are gathered and processed as the RAS moves dynamically through the regulation and interacts with the user. Third, results of the analysis are compiled and presented to the user. The following sections describe in detail the three stages of the compliance assistance process.

3.2.1 XML Regulation Verification

The RAS performs two verification checks on an XML regulation before the regulation is used for compliance checking. First, the XML regulation is checked against a regulation DTD (Document Type Definition), which defines the valid structures for an XML regulation (Navarro et.al. 2000). The verification with the DTD primarily provides a "grammatical" check to ensure that the structure of the regulation can be parsed and interpreted. The second step is to verify that all the logic rules contained in logicSentence elements are consistent. The initial check for problems with the logic rules is important. The RAS system performs compliance checks by identifying logical contradictions between user input and regulation rules. If the regulation rules themselves contain a contradiction, compliance checking cannot proceed.

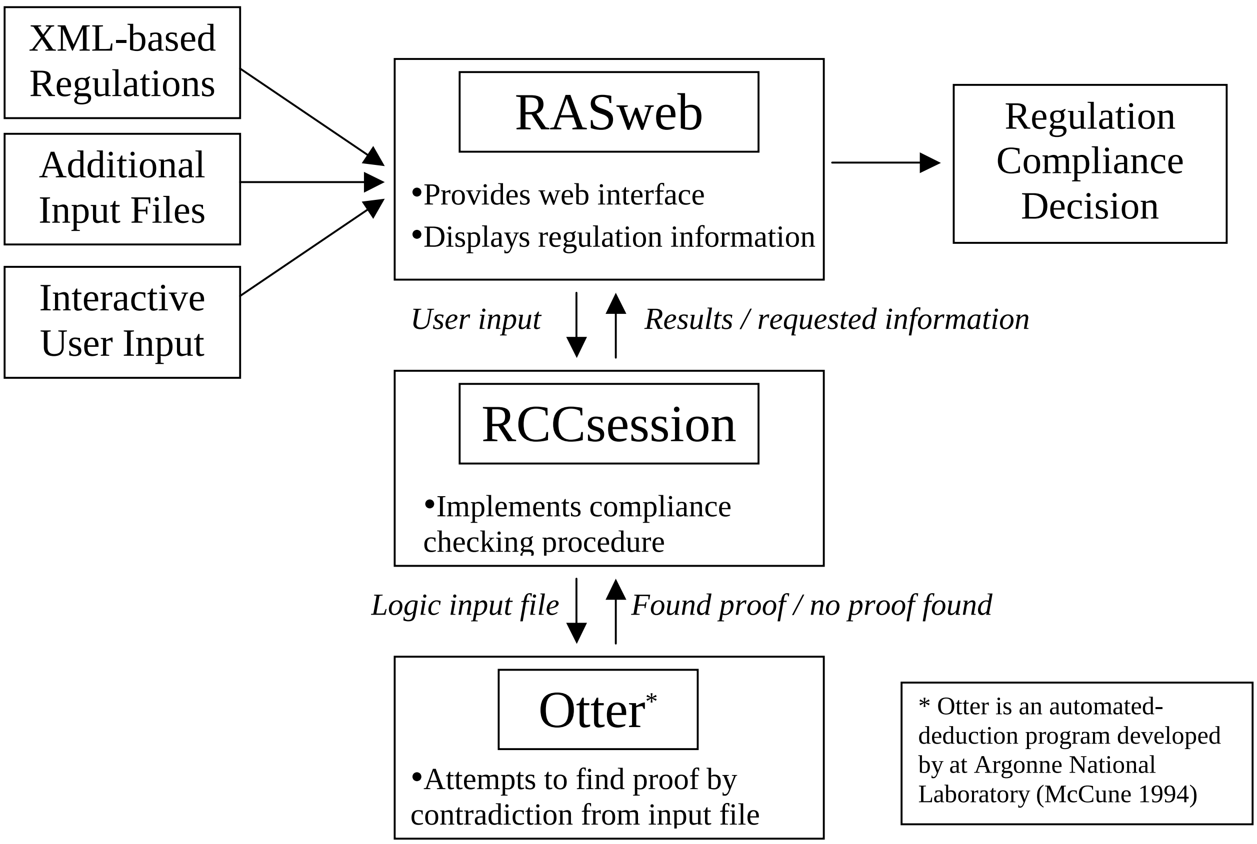


Figure 9. Organization of regulation assistance system

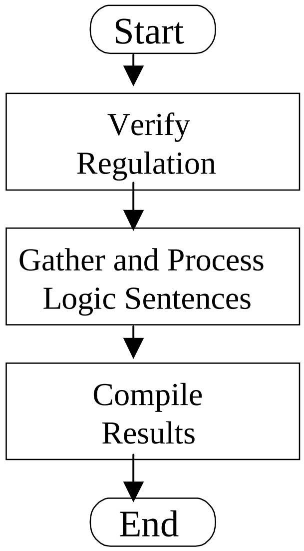


Figure 10. Regulation compliance process

To check for consistency of the logicSentence elements, the system extracts all the logicSentence elements from the target regulation and sends them to the theorem prover, Otter. If the theorem prover does not find a contradiction in the logic sentences within a given time period, the logic sentences are assumed to be consistent. This check attempts to ensure that the set of logic rules embedded in the XML regulation do not contain a contradiction. The initial check for contradictions in regulation rules does not guarantee that there are no contradictions in the rules, since Otter is not guaranteed to find a contradiction if one exists. In our experience, however, this initial logic check has been fairly robust.

3.2.2 Gathering and Processing Logic Sentences

Given an initial provision selected for compliance checking, related provisions are gathered and processed using a modified depth-first tree traversal of the XML structure. The procedure deviates from depth-first processing as control elements are encountered. The initially selected provision can be anywhere in the XML regulation tree. A provisions-to-process (PTP) stack maintains a list of regulation provisions that need to be investigated, and an already-processed-provisions (APP) list keeps track of the provisions for which processing is complete.

The compliance checking process is basically a process of gathering logic sentences from the regulation rules, control elements, and user responses and doing proofs to check for contradictions.

Figure 11 shows the flowchart for gathering and processing logic sentences. The system starts from the top provision in the PTP stack. Any logic rules from that provision are noted, and any control elements found are also processed. If there is no question associated with the provision or its sub-provisions, the current provision is moved to the APP list and the next provision on the PTP stack is processed. If there is a question (i.e., a logicOption element) associated with the provision, the user is asked for an answer to the question. The XML elements, both logic and control elements, associated with the user's answer are then processed before the system continues to the next question. If the PTP stack becomes empty or a contradiction is found in the logic sentences, this stage of processing is complete. The details for gathering and processing logic sentences are described below.

3.2.2.1 Processing logicOption Elements

The procedure for identifying contradictions between the user inputs and the regulation rules is shown in Figure 12. The logicOption elements provide a mapping from the user responses to the control elements that direct further processing and the logic sentences that can be verified against logic rules in the regulation. After each question is answered, the logic associated with the selected answer is recorded and any control elements associated with the answer are processed. The theorem prover is sent all the regulation rules encountered during processing; these include all the logic sentences selected by the users in response to questions, and the logic sentences stating that the provisions with which the compliance check was initiated must be satisfied.

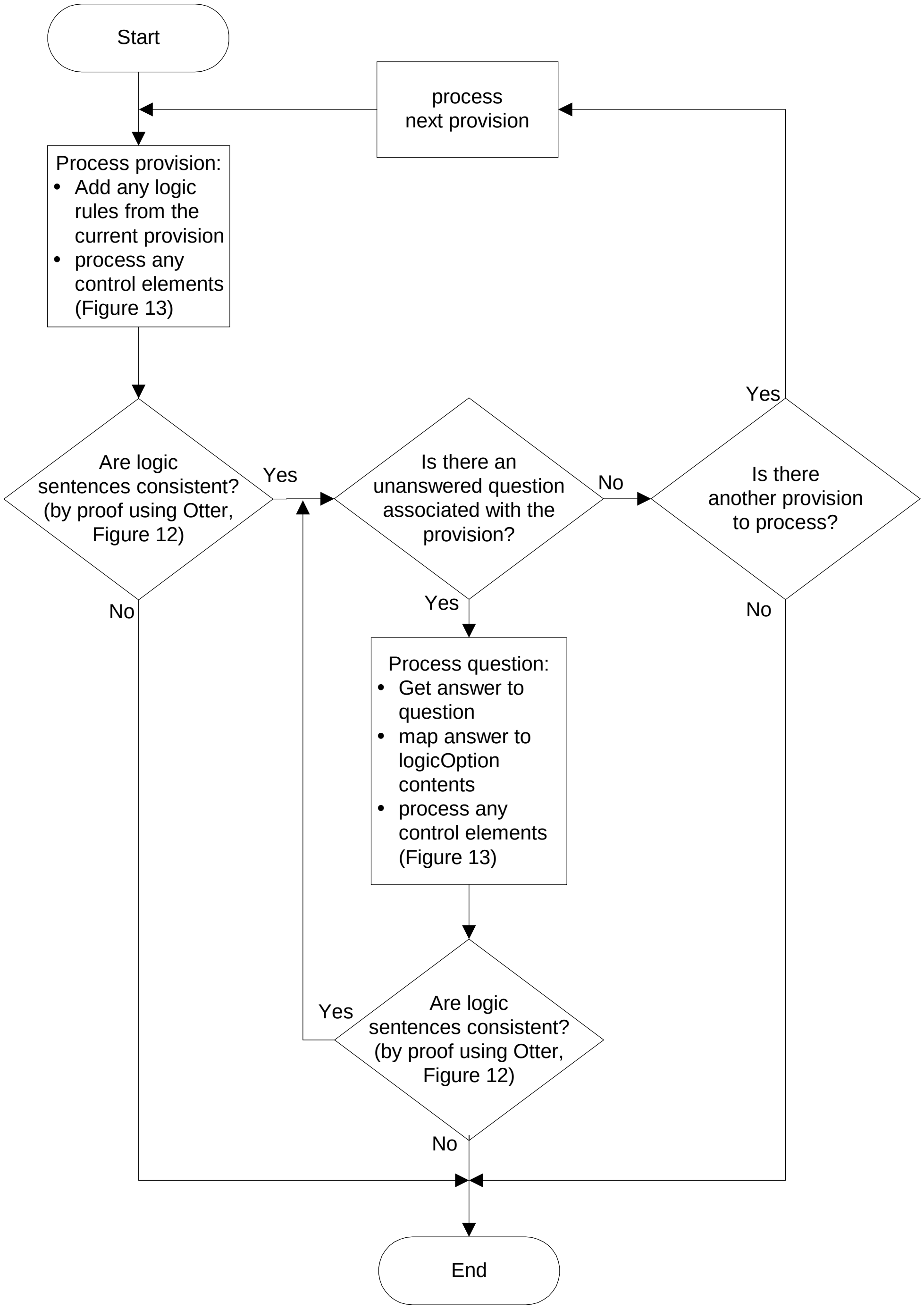


Figure 11. Gathering and processing logic sentences

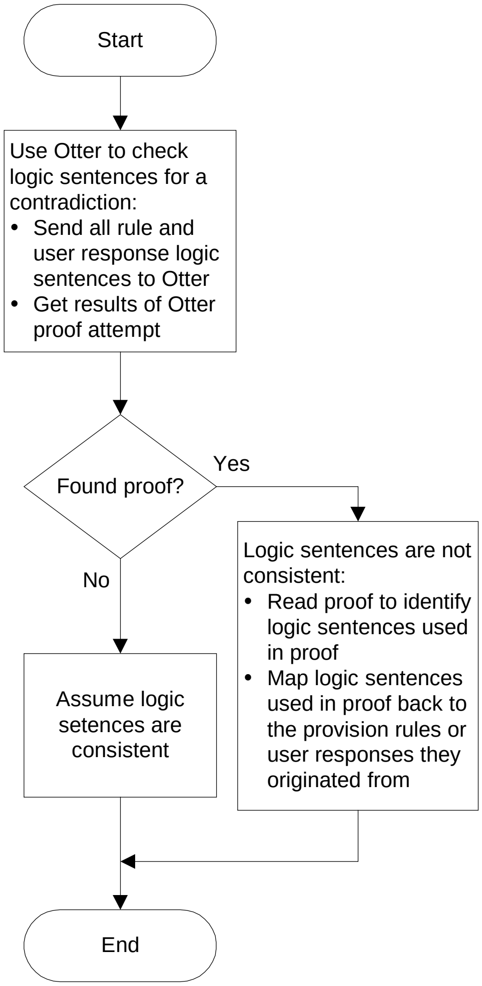


Figure 12. Processing FOPC with Otter

After Otter attempts to find a proof, the RAS checks to see if the theorem prover has been able to find a contradiction in the input logic sentences. If no proof has been found, the logic sentences are assumed to be consistent. If a proof has been found, the proof steps are read to find the input logic sentences that contributed to the contradiction. These input logic sentences are then mapped back to the provision rules or the user responses from which they originated. This allows the system to identify what is contributing to the logical contradiction (i.e., non-compliance with the regulation).

Answers to the questions contained in the logicOption elements are recorded in a log file. This file enables the system to automatically process questions that have been answered in the past. This log file of answers also forms a detailed audit trail that can be provided to the user.

3.2.2.2 Processing Control Elements

Figure 13 shows how the two types of control elements, simple and conditional, are processed. First, rules are followed to update the PTP stack and APP list according to any simple control statements. Second, the system iterates through each conditional control statement and attempts to prove that the conditions of the control statement are satisfied.

The processing method and the effects of the three simple control elements on the PTP stack and APP list are shown in Figures 14-16. For the goto control element, examples of the initial PTP stack are shown on the left and the resulting stacks after taking the goto elements into account are shown on the right side of Figure 14. In the simplest case (1), the goto element adds the new provision specified to the PTP stack. In case (2), adding a provision to the stack is ignored because only a single call to a particular regulation provision may be in the PTP stack at a time to prevent infinite loops. Case (3) illustrates the idea that any processed provisions in the APP list cannot be added to the PTP stack. Case (4) demonstrates that even if the system is processing a sub-provision of the top PTP provision, the goto element operates similar to the other cases (for example, adding the new provision to the PTP stack).

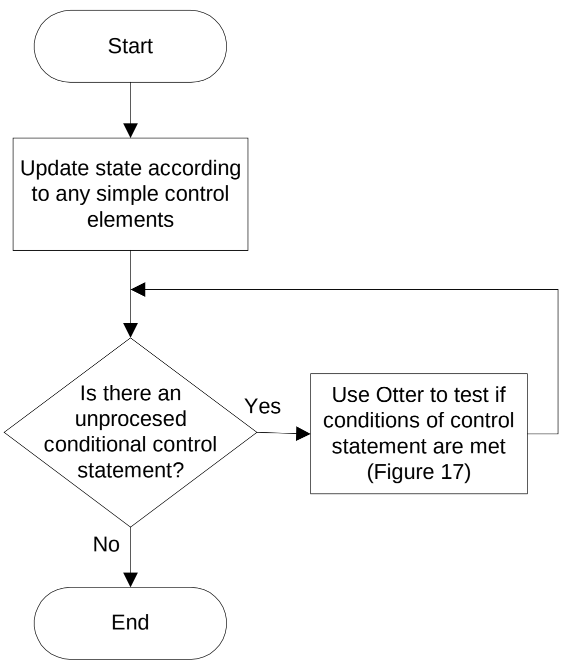


Figure 13. Processing control elements

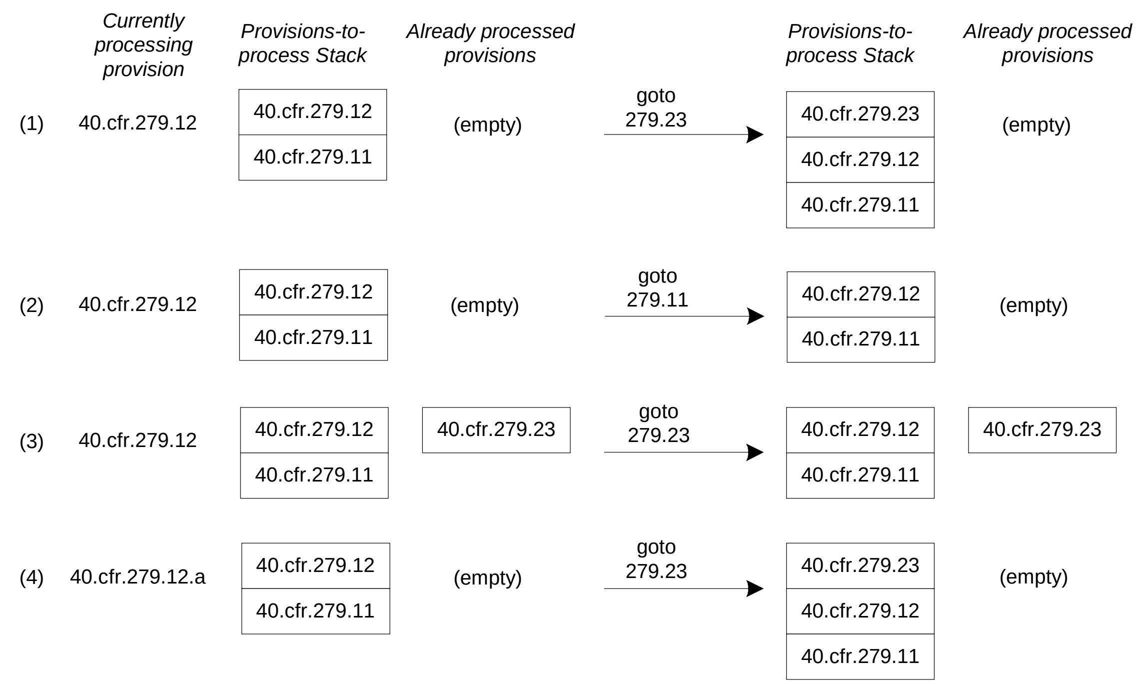


Figure 14. The goto element

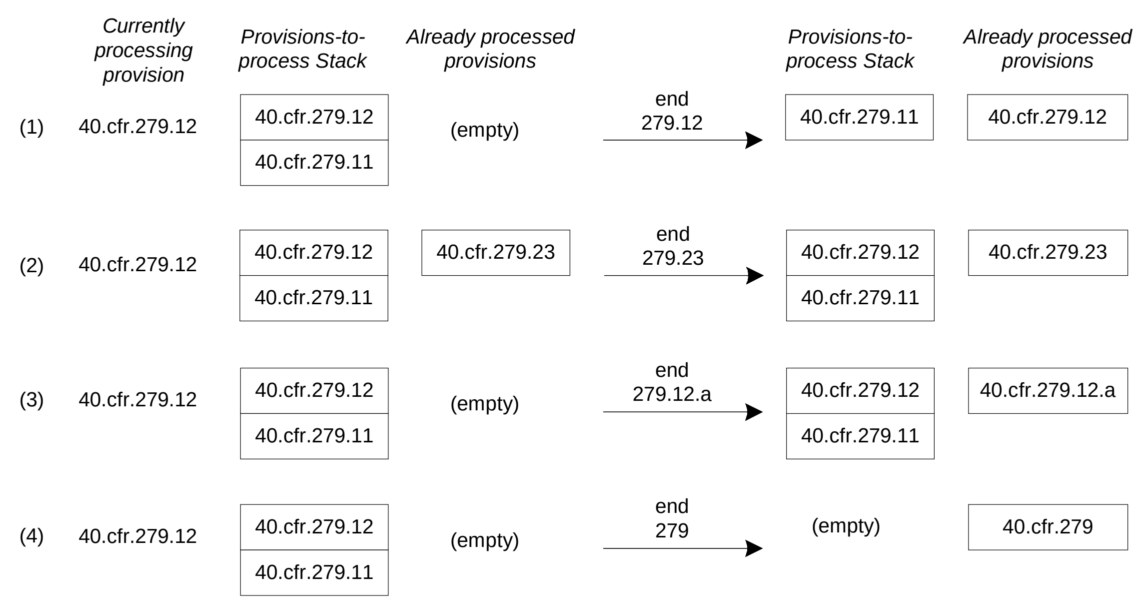


Figure 15. The end element

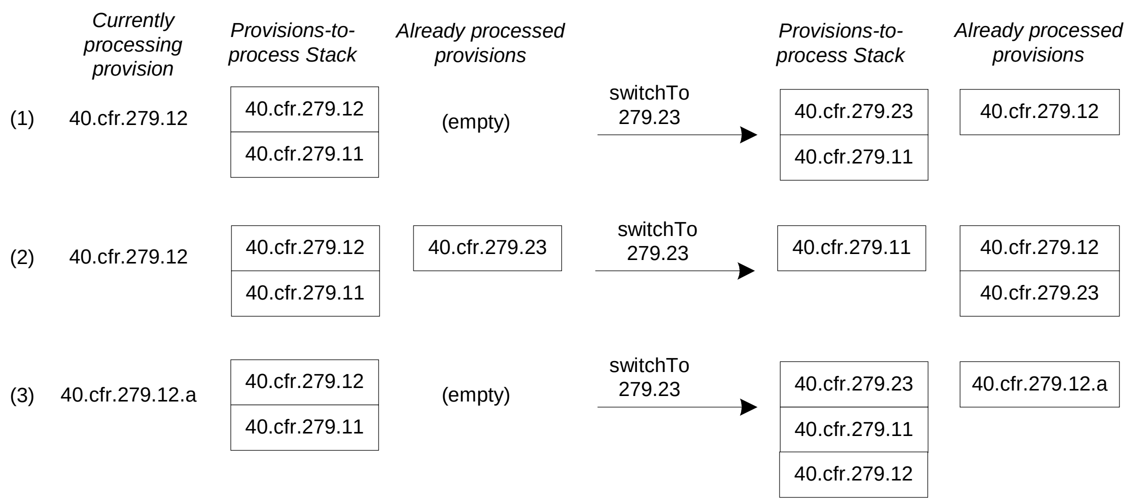


Figure 16. The switchTo element

The processing rules for the end and switchTo control elements are shown in Figure 15 and Figure 16, respectively. The basic effect of the end control element is that the targeted provision is removed from the PTP stack if it exists there, and added to the APP list. The basic effect of the switchTo control element is that the top provision is removed from the PTP stack and added to the APP list, and the provision specified by the switchTo element is added to the PTP stack. The switchTo element is provided for convenience, since it has the same effect as a goto element combined with an implied end element for the current provision.

Processing the logic-based conditional control elements requires the use a FOPC theorem prover to check if the antecedents of the implications are satisfied. The process is shown in Figure 17. For each conditional control statement, Otter is sent all current logic rule and answer sentences, along with a logic sentence negating the target of the logic-based control statement. The negated target of the conditional control statement is necessary because the theorem prover constructs proofs by contradiction. If Otter finds a contradiction, the control statement is executed. If the statement is of the "provApplies" variety, the targeted provision is added to the PTP stack. If the statement is of the "provDoesNotApply" variety, the targeted provision is added to the APP list and removed from the PTP stack if it is located there. The targeted provision may not necessarily be in the PTP stack, since it may not have been previously added to the stack.

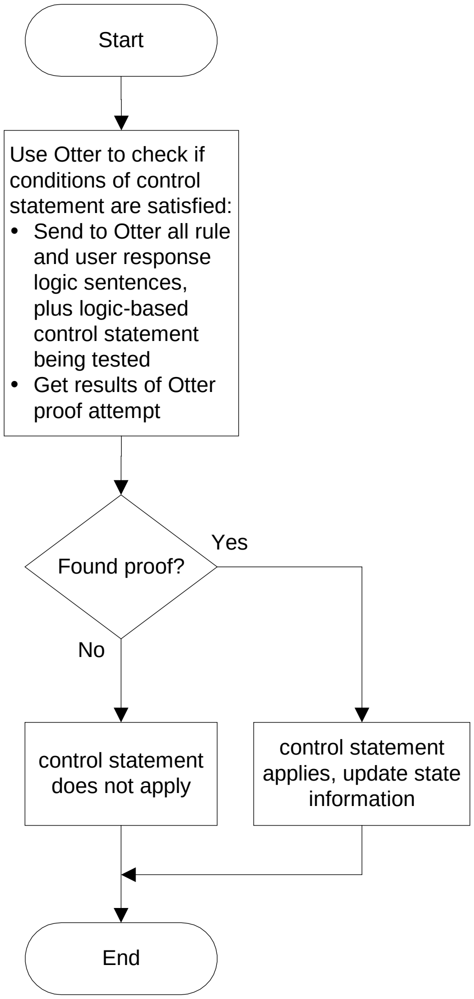


Figure 17. Processing logic-based control statements

3.2.3 Compilation of Results

The questioning procedure terminates when either a logical contradiction is found or the PTP stack is empty. When the questioning procedure ends due to a logical contradiction being found, the system returns a result stating that there is a compliance problem and a detailed report is provided for the user to help identify the problem. All the questions, answers and relevant provisions that contributed to the logical contradiction are displayed for the user. An example screen shot of the system results is shown in Figure 18.

When the questioning procedure ends due to an empty PTP stack, the system has examined all relevant logic sentences and failed to find any logical contradictions (i.e., the system is unable to prove

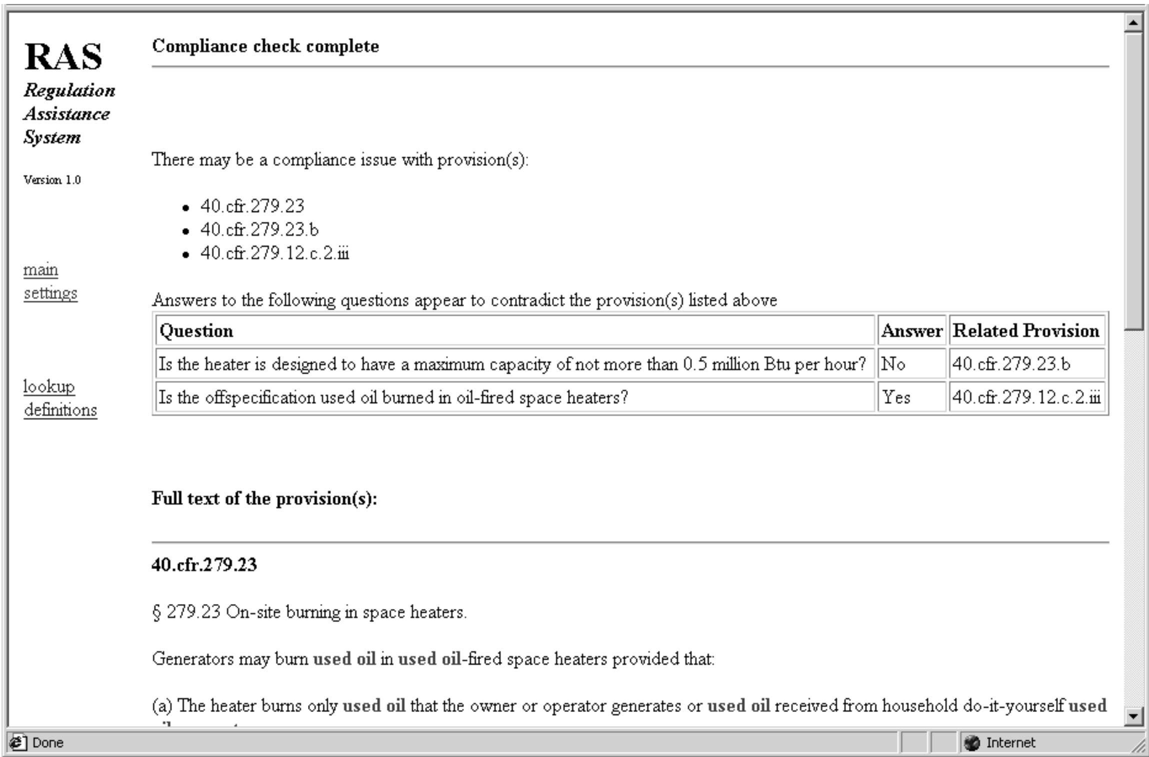


Figure 18. Compliance summary with questions contributing to non-compliance shown

noncompliance). Note that this is different from proving that the user is in compliance with the regulation. The procedure attempts to show noncompliance with the regulation, and when the procedure fails to show noncompliance the system returns a result stating that it appears the user is in compliance with the regulation.

4 Applications

The regulation assistance system (RAS) has been built as a demonstration platform to implement the compliance checking procedure and to illustrate some of the tools achievable with the metadata in the XML regulation framework. The RAS has been written as a Java servlet, a java-based program that is designed to run on servers and is similar in usage to CGI (Common Gateway Interface) programs (Callaway 2001). The RAS is run as a web application by Tomcat, a java-based application server being produced by the Apache Jakarta Project (Brittain and Darwin 2003) These examples show the use of the RAS for a compliance checking session. and illustrate the possibility of linking the RAS with an online guidance system.

4.1 Compliance checking session

Figure 19 shows the main menu of the RAS, where the user can enter the regulation provision or provisions to be checked. Then, the system asks the user a series of questions, halting when it makes a compliance decision. Figure 20 is a screen shot of a compliance session in progress. The text for the current section of the regulation is shown while the system is asking the user questions, with the exact provision shown in bold. In addition, explicit definitions of terms and acronyms are incorporated in the user interface by highlighting words with definitions, and providing pop-up definition or acronym explanations when a user moves the mouse over the highlighted terms. The RAS also allows the browsing of cross-referenced regulation provisions to make reading the regulation less cumbersome. An example of a reference link is shown in Figure 20 with the underlined link following Subsection 40 CFR 279.71(b).

The regulation assistance system assists the user in locating supplementary documents such as guidance documents, letters of interpretations, and administrative decisions by using the "concept" elements in the XML regulation to link regulation provisions to the document repository. By identifying documents in the document repository that share "concepts" in common with a particular regulation provision, supplementary information that is relevant to that provision can be identified. An example of this feature is shown in Figure 20 and Figure 21. Figure 20 shows the concepts from the XML regulation, which function as predefined search terms, linking to the document repository. Figure 21 shows how concept searches can lead into the document repository, and from there users can locate relevant supplementary documents.

In addition to the answers for logicOption elements, the system also includes an optional answer for "I don't know" (see Figure 20). With this option, the system forks the compliance-checking process along all possible answers to the question. This allows the user to explore all available questions and answers for the compliance session. Once the user has answered all the questions for all the compliance checking cases, the results for each case are presented as shown in Figure 22.

Upon completing a compliance check, a user may view and download a log file of the compliance checking session as shown in Figure 23. This feature is valuable for record keeping or when revisiting the regulations at a later date. Uploading log files allows users to check for compliance against regulations that have been modified since the previous compliance check. Log files may be modified and resubmitted to reflect changing operations or allow checking of different scenarios. Modifications to the log file are made by simply removing the answers that a user does not wish to keep, as shown in Figure 24.

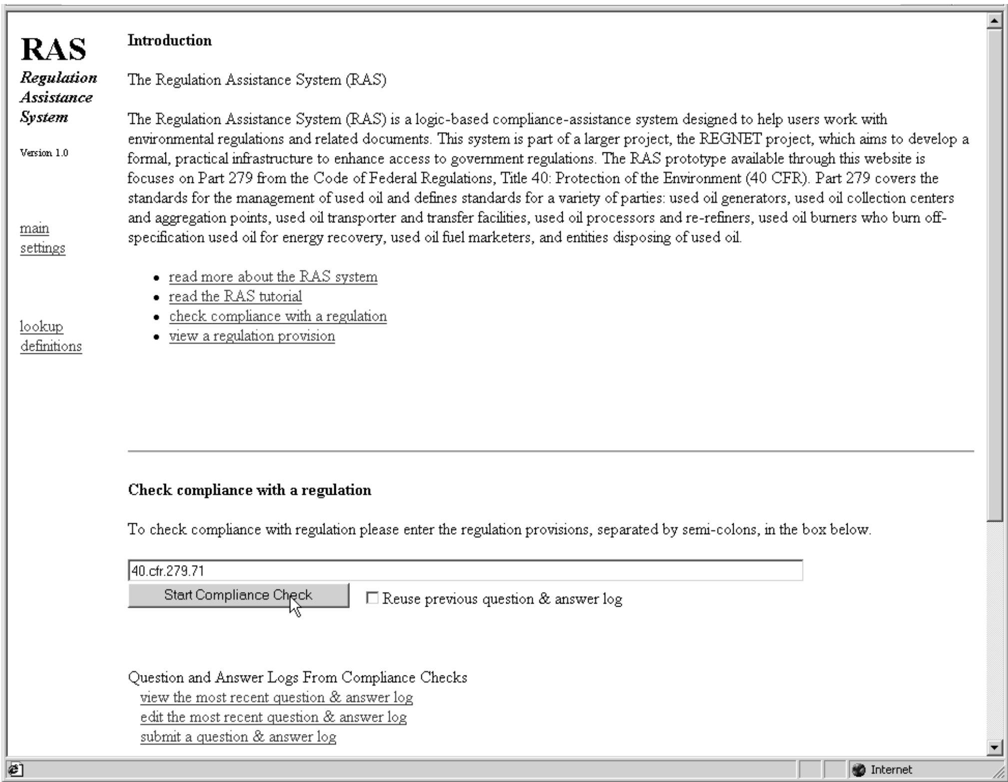


Figure 19. Main menu of the regulation assistance system

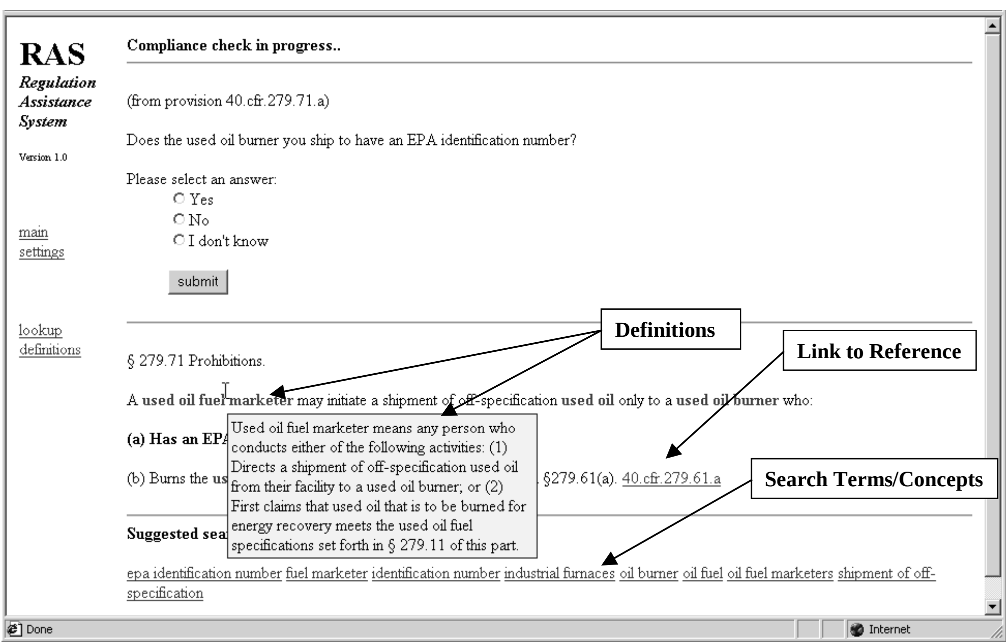


Figure 20. Compliance assistance check in progress

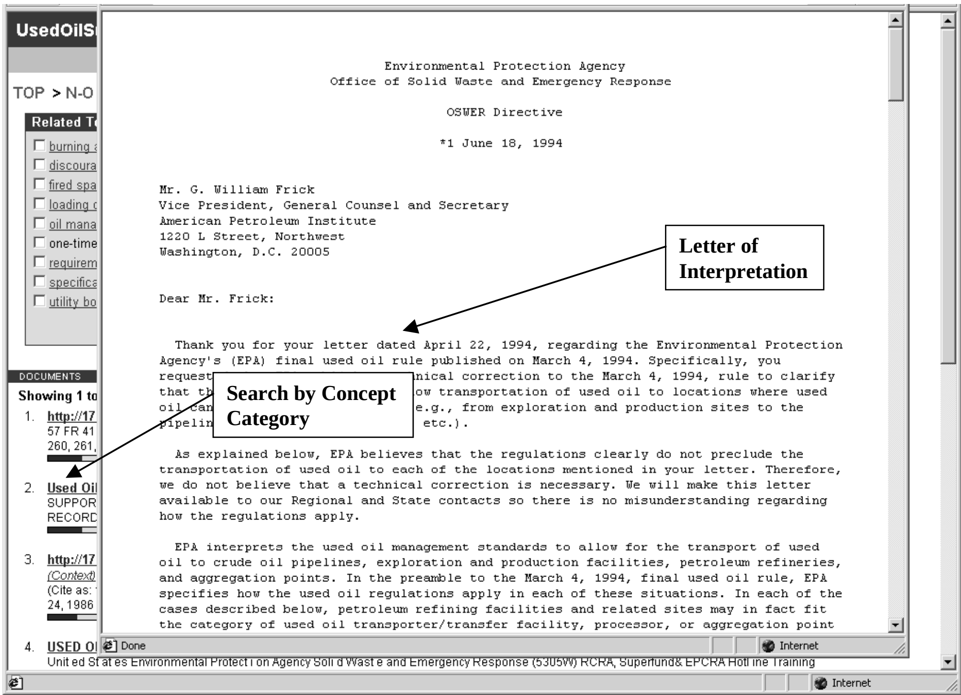


Figure 21. Accessing documents in the document repository

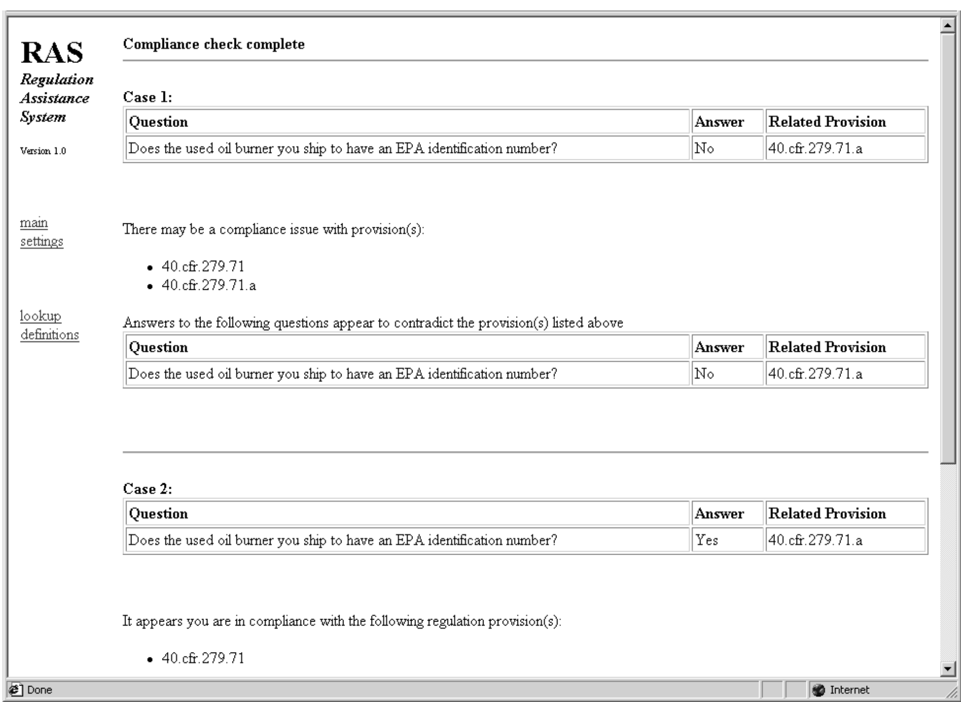
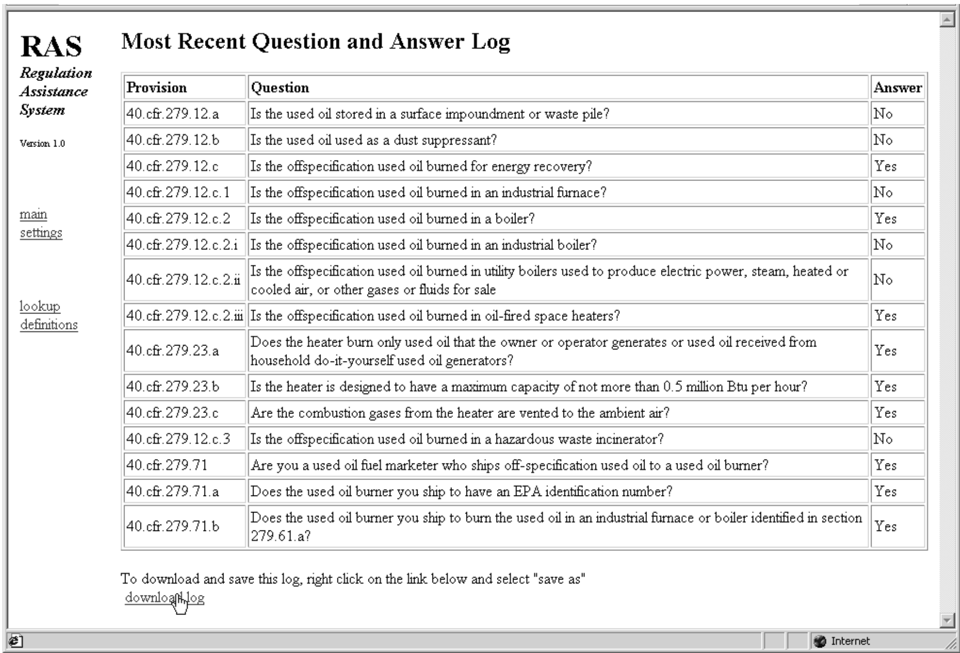


Figure 22. Example with multiple answers during compliance checking



RAS
Regulation Assistance System
Version 1.0

main
settings

lookup
definitions

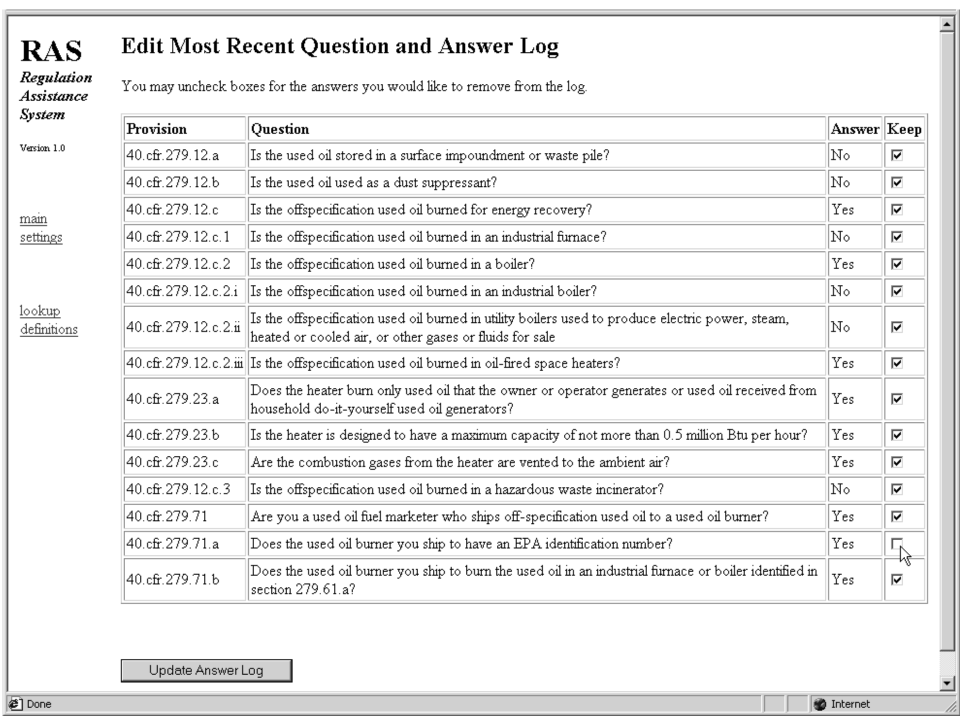
Most Recent Question and Answer Log

Provision	Question	Answer
40.cfr.279.12.a	Is the used oil stored in a surface impoundment or waste pile?	No
40.cfr.279.12.b	Is the used oil used as a dust suppressant?	No
40.cfr.279.12.c	Is the offspecification used oil burned for energy recovery?	Yes
40.cfr.279.12.c.1	Is the offspecification used oil burned in an industrial furnace?	No
40.cfr.279.12.c.2	Is the offspecification used oil burned in a boiler?	Yes
40.cfr.279.12.c.2.i	Is the offspecification used oil burned in an industrial boiler?	No
40.cfr.279.12.c.2.ii	Is the offspecification used oil burned in utility boilers used to produce electric power, steam, heated or cooled air, or other gases or fluids for sale	No
40.cfr.279.12.c.2.iii	Is the offspecification used oil burned in oil-fired space heaters?	Yes
40.cfr.279.23.a	Does the heater burn only used oil that the owner or operator generates or used oil received from household do-it-yourself used oil generators?	Yes
40.cfr.279.23.b	Is the heater is designed to have a maximum capacity of not more than 0.5 million Btu per hour?	Yes
40.cfr.279.23.c	Are the combustion gases from the heater are vented to the ambient air?	Yes
40.cfr.279.12.c.3	Is the offspecification used oil burned in a hazardous waste incinerator?	No
40.cfr.279.71	Are you a used oil fuel marketer who ships off-specification used oil to a used oil burner?	Yes
40.cfr.279.71.a	Does the used oil burner you ship to have an EPA identification number?	Yes
40.cfr.279.71.b	Does the used oil burner you ship to burn the used oil in an industrial furnace or boiler identified in section 279.61.a?	Yes

To download and save this log, right click on the link below and select "save as"
download log

Internet

Figure 23. A log of compliance assistance session



RAS
Regulation Assistance System
Version 1.0

main
settings

lookup
definitions

Edit Most Recent Question and Answer Log

You may uncheck boxes for the answers you would like to remove from the log.

Provision	Question	Answer	Keep
40.cfr.279.12.a	Is the used oil stored in a surface impoundment or waste pile?	No	☑
40.cfr.279.12.b	Is the used oil used as a dust suppressant?	No	☑
40.cfr.279.12.c	Is the offspecification used oil burned for energy recovery?	Yes	☑
40.cfr.279.12.c.1	Is the offspecification used oil burned in an industrial furnace?	No	☑
40.cfr.279.12.c.2	Is the offspecification used oil burned in a boiler?	Yes	☑
40.cfr.279.12.c.2.i	Is the offspecification used oil burned in an industrial boiler?	No	☑
40.cfr.279.12.c.2.ii	Is the offspecification used oil burned in utility boilers used to produce electric power, steam, heated or cooled air, or other gases or fluids for sale	No	☑
40.cfr.279.12.c.2.iii	Is the offspecification used oil burned in oil-fired space heaters?	Yes	☑
40.cfr.279.23.a	Does the heater burn only used oil that the owner or operator generates or used oil received from household do-it-yourself used oil generators?	Yes	☑
40.cfr.279.23.b	Is the heater is designed to have a maximum capacity of not more than 0.5 million Btu per hour?	Yes	☑
40.cfr.279.23.c	Are the combustion gases from the heater are vented to the ambient air?	Yes	☑
40.cfr.279.12.c.3	Is the offspecification used oil burned in a hazardous waste incinerator?	No	☑
40.cfr.279.71	Are you a used oil fuel marketer who ships off-specification used oil to a used oil burner?	Yes	☑
40.cfr.279.71.a	Does the used oil burner you ship to have an EPA identification number?	Yes	☐
40.cfr.279.71.b	Does the used oil burner you ship to burn the used oil in an industrial furnace or boiler identified in section 279.61.a?	Yes	☑

Update Answer Log

Done
Internet

Figure 24. Editing compliance assistance log

4.2 Linking Online Guidance Systems with the RAS

The regulation assistance system primarily addresses the problem of helping to determine whether one is in compliance with a regulation by guiding the users through the regulation. The RAS can also be used as a component to be linked to by other systems. Towards that end, the RAS is designed such that it can initiate compliance checks at any point within a regulation, and a compliance check can be started by connecting to the RAS with a target regulation encoded in a web browser's URL.

To demonstrate, a sample online guide is built for vehicle maintenance shops. The online guide is adapted from a paper-based guide developed by the New York State Department of Environmental Conservation Pollution Prevention Unit (2002). Our adaptation is for demonstration purposes only since the original guide provides state regulation references while our online guide links users to federal regulations analogous to the state requirements. In the case of used oil regulations the New York state regulations are similar to the federal regulations, so linking to federal regulations adequately illustrates the functionality possible with the system.

The vehicle maintenance guide explains in plain language why vehicle maintenance shops are regulated, and how the vehicle maintenance shops should follow the regulations. The guide then lists a number of common materials and activities used by vehicle maintenance shops in the course of business. Each of these materials or activities has a web page dedicated to explain in plain language the regulatory requirements governing the material or activity. The original paper-based guide explains general requirements and then references applicable regulations for more detail. This creates a problem, because when readers are referred to the regulation, they are back to the original dilemma that the guide is attempting to address; the problem of dealing with all the issues associated with finding, working with, and interpreting regulations. The online adaptation provides a solution to the reference problem in the form of an additional feature that links references to the regulation assistance system. These links enable users to click on referenced regulations, which will connect them to the RAS to check for compliance.

Figure 25 through Figure 27 illustrate the link between the vehicle maintenance shop online guide and the regulation assistance system. Figure 25 shows the web page for the vehicle maintenance shop online guide, from which users may access information on specific materials or processes, like used oil. Selecting the used oil link brings the user to the web page illustrated in Figure 26, which shows the regulatory requirements for used oil. Note the reference in Figure 26 to a regulatory provision, 40 CFR 279.23, which is used as a link to the regulation assistance system. Figure 27 shows the RAS system, as

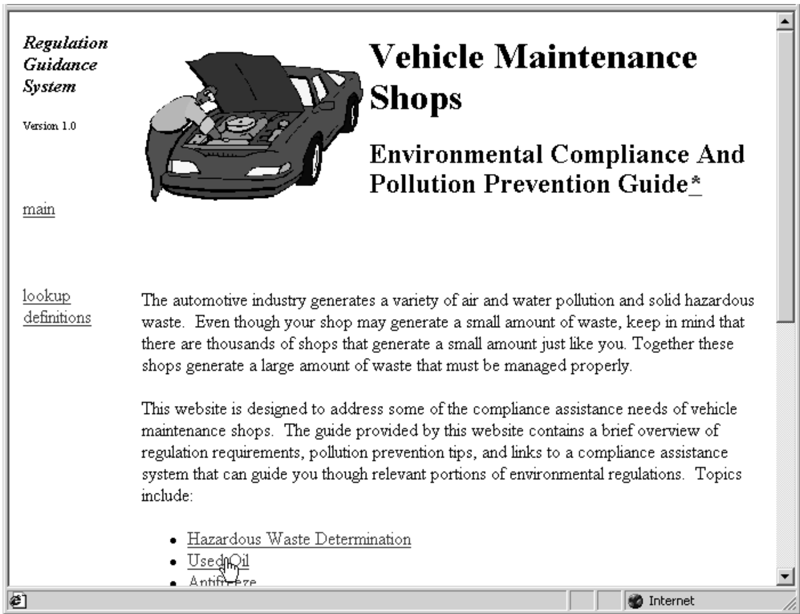


Figure 25. Vehicle maintenance shop compliance guide

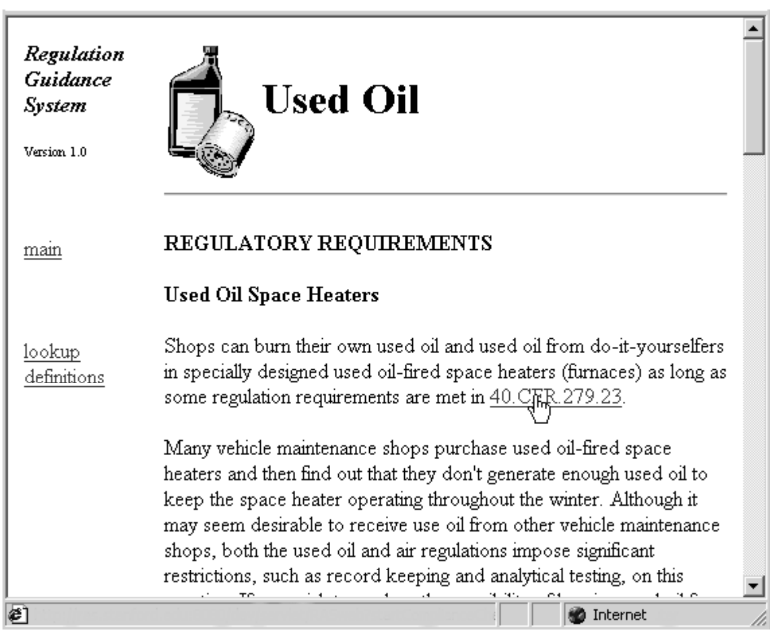


Figure 26. Compliance guide for used oil

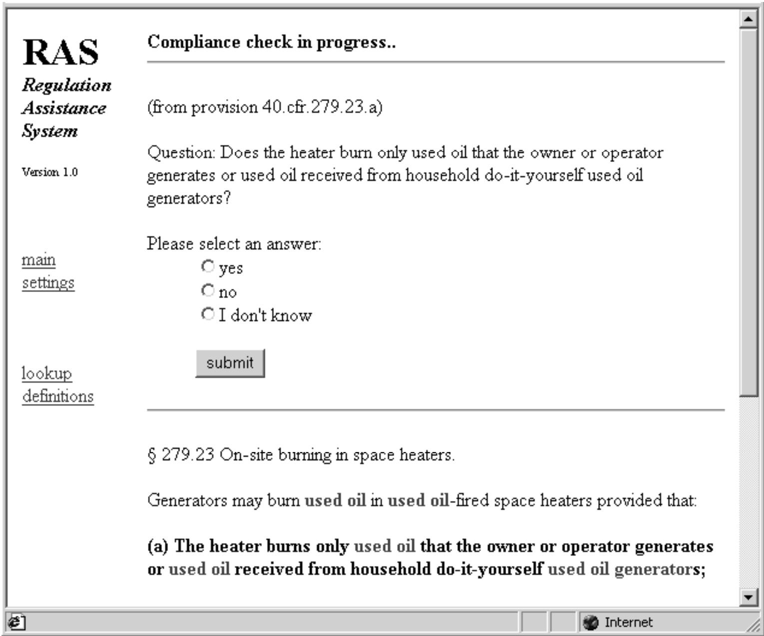


Figure 27. Compliance guide linked to RAS

accessed from the used oil web page of the vehicle maintenance shop online guide. From the RAS system users can check for compliance with the referenced used oil regulation provision or connect to the document repository to look for related supplementary documents.

5 Related Research

Representation of regulations and laws has been an active research area for decades. There has been a great deal of work done on building expert systems for law (Sergot et.al. 1986, Wahlgren 1992, Zeleznikow and Hunter 1994). Bench-Capon provided a review on the applications of knowledge-based systems for legal applications, particularly the research and development efforts related to the Alvey DHSS Demonstrator project in the U.K. (Bench-Capon 1991). The reference includes a large number of citations that appeared before 1990 that are related to logic and rule based approaches and their application in legal systems. Much of the earlier work in IT and law focused on building systems to optimize decisions with respect to laws, particularly tax law (McCarty 1977). Logic-based approaches have also been applied to engineering standards processing (Kiliccote 1996, Kiliccote and Garrett 1998, Yabuki 1992). Some of the recent work has focused on investigations into case-based reasoning and information retrieval (Brüninghaus and Ashley 1997). Methodologies for tailoring legal documents to users' needs have also been studied (Royles and Bench-Capon 1998, Royles 2000). While legal knowledge representation and reasoning has been an active research topic (ICAIL 1999, 2001, 2003), an integrated approach covering the management of regulations, efficient access and retrieval of documents and tools for compliance checking is missing.

The past thirty years have seen significant advancements in theorem-proving technology (Wos and Pieper 1999). Research for new formalisms and specialized logics continue to improve reasoning speed and non-monotonic reasoning capabilities (Greiner et.al. 2001, Shanahan 1997). FOPC does not have the expressive power to deal with issues of open texture, deontic modality, or subjunctive conditionals, which are active areas of research (Jones and Sergot 1993, Sanders 1991). Even though we use a simplified representation of the regulation rules in this work, logic and other metadata may be useful for a variety of systems (Lauritsen 1993).

Two research projects in particular are closely related to the work presented in this paper. Royles (2000) wrote his Ph.D. thesis on the intelligent presentation and tailoring of online legal information. A prototype implementation was built to provide private consultations with users to help them identify relevant benefits they might be able to collect from the government. Royles' work provides important guidance for how many of the privacy questions that might arise from work in this paper could be addressed, and provides a model for how a tiered implementation of the compliance system might be designed.

Wang (2003) discussed in his thesis the development of an integrated and distributed information management infrastructure to support hazardous waste compliance, research work that was a precursor to the work presented in this paper, deals with the information organization of regulations and the issues of information interoperability for the compliance process. These are important issues in the design of any regulation compliance assistance system.

6 Summary and Discussion

The goal of the REGNET research project is to develop an infrastructure for regulatory information management and compliance assistance. This paper describes our research on developing a compliance assistance infrastructure that builds upon an XML-based regulation framework. There are two distinct features worth mentioning.

Annotating XML regulations with logic elements and processing them in the manner described in this paper has a performance advantage over simply building a large knowledge base of logic sentences. The primary advantage of the approach described is that the number of logic sentences that need to be handled by the reasoning subsystem (i.e., Otter) is reduced. The XML structure allows the system to properly scope the meta-data and reduce the amount of extraneous data passed to the reasoning system. Only the logic and control processing metadata necessary for the compliance checking are acquired and dynamically loaded into the reasoning system. This is important because doing logic proofs is computationally intense, and reducing the number of extraneous logic sentences reduces the processing time for proofs and increases the complexity of problems that can be handled.

Online regulation guides located anywhere can build upon the compliance-checking capabilities of the regulation assistance system. Many different compliance guides provided by regulators, industry trade groups, or commercial third party assistance providers could build upon this design by developing online plain language compliance guides linked to regulation assistance systems. An individual attempting to comply with regulatory requirements could identify a relevant online guide addressing the appropriate industry focus for his or her situation. The online guide could then refer the user to the relevant parts of applicable regulations by using hyperlinks to a regulation assistance system. This design allows many different online guides to all refer back to a single regulation assistance system.

This paper has described current work on building an initial document repository, XML structure for regulations, and a regulation assistance system. There are many future research questions related to regulation management system. These include: How can automated tools be built to help entities find information on state and federal laws, as well as identify sources of assistance with compliance questions and problems? How can the XML regulation structure be extended, particularly to allow other logic formalisms and more advanced annotation with legal interpretations? Can more advanced forms of logic be incorporated to more precisely represent the regulation? How will multiple, domain specific, regulation assistance systems interoperate? How can security and privacy be provided when using a compliance assistance system?

Acknowledgments

This research project is sponsored by the National Science Foundation, Contract Numbers EIA-9983368 and EIA-0085998. The authors would like to acknowledge an equipment grant from Intel Corporation and the support by Semio Corporation in providing the software for this research. The authors would like to thank Professors Gio Wiederhold of the Computer Science Department, Barton Thompson of the Law School, and Jim Leckie of the Department of Civil and Environmental Engineering for their valuable suggestions in this project. Last but not least, the authors would like to acknowledge the contributions by the REGNET research team members, Charles Heenan, Haoyi Wang, Gloria Lau and Jie Wang, in the Engineering Informatics Group.

References

Bench-Capon, T. J. M. (1991), *Knowledge Based Systems and Legal Applications,* The APIC Series 36, Academic Press.

Botkin, A. (2002), "Wizards, Advisors and Websites, Oh My! Interactive Electronic Tools for Compliance Assistance," presented at the *National Compliance Assistance Providers Forum*, U.S. Environmental Protection Agency and Texas Commission on Environmental Quality, San Antonio, TX . (see http://www.epa.gov/compliance/resources/briefings/assistance/stakeholders/electronic.pdf.)

Brittain, J. and Darwin, I. F. (2003), *Tomcat: The Definitive Guide*, O'Reilly & Associates.

Brüninghaus, S. and Ashley, K. D. (1997), "Finding Factors: Learning to Classify Case Opinions Under Abstract Fact Categories," *Proceedings of Sixth International Conference on Artificial Intelligence and Law*, Melbourne, Australia, ACM Press, pp. 123-131.

Van Wert, J.M. (2002), "Business Compliance One-Stop: Creating a Single Face of Government," presented at the *National Compliance Assistance Providers Forum*, U.S. Environmental Protection Agency and Texas Commission on Environmental Quality San Antonio, TX. (see http://www.epa.gov/compliance/resources/briefings/assistance/stakeholders/initiative.pdf.)

Callaway, D. R. (2001), *Inside Servlets: Server-Side Programming for the Java Platform*, 2nd Edition, Addison-Wesley.

Genesereth, M.(1992), *Knowledge Interchange Format*, Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, CA.

Greiner, R., Darken, C., and Santoso, N. I. (2001), "Efficient Reasoning," *ACM Computing Surveys*, Volume 33, Issue 1, pp. 1-30.

Heffron, F. A. and McFeeley, N. (1983), *The Administrative Regulatory Process*, Longman.

Jones, A. J. I. and Sergot, M. (1993), "On the Characterisation of Law and Computer Systems: the Normative Systems Perspective," in *Deontic Logic in Computer Science: Normative System Specification*, J.-J.C. Meyer and R.J. Wieringa, Editors. John Wiley Publ. Co., pp. 275-307.

Kerrigan, S. L. (2003), *A Software Infrastructure for Regulatory Information Management and Compliance Assistance*, Ph.D. Thesis, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA.

Kiliccote, H. (1996), *A Standards Processing Framework*, Ph.D. Thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA..

Kiliccote, H. and J. H. Garrett, Jr. (1998), "Standards Modeling Language," *Journal of Computing in Civil Engineering*, Vol. 2, No. 9, pp. 129-135.

Lauritsen, M. (1993), "Knowing Documents," *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, Amsterdam, The Netherlands, ACM Press, pp. 184-191.

McCarty, T. (1977), "Reflections on Taxman: An Experiment in Artificial Intelligence and Legal Reasoning," *Harvard Law Review*, Volume 90, pp. 837-893.

McCune, W. W. (1994), "Otter 3.0 Reference Manual and Guide," ANL-94/6, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL.

National Compliance Assistance Providers Forum (2002), U.S. Environmental Protection Agency and Texas Commission on Environmental Quality, San Antonio, TX.

Navarro, A., White, C., and Burman, L. (2000), *Mastering XML*, SYBEX Inc..

New York State Department of Environmental Conservation Pollution Prevention Unit (2002), *Environmental Compliance And Pollution Prevention Guide for Vehicle Maintenance Shops*.

ICAIL (1999), *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, Oslo, Norway, ACM Press.

ICAIL (2001), *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, St. Louis, U.S., ACM Press.

ICAIL (2003), *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, Edinburgh, Scotland, UK, ACM Press.

Royles, C. A. and Bench-Capon, T. J. M. (1998), "Dynamic Tailoring of Law Related Documents to User Needs," *Proceedings of 9th International Workshop on Database and Expert System Applications*, Vienna, Austria, IEEE, pp. 609-613.

Royles, C. A. (2000), *Intelligent Presentation and Tailoring of Online Legal Information*, Ph.D. Thesis in Department of Computer Science, University of Liverpool, Liverpool, U.K..

Sanders, K. E. (1991), "Representing and Reasoning About Open-Textured Predicates," *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford, England, ACM Press, pp. 137-144.

Sergot, M. J., Sadri, F., Kowalski, R. A., Kriwaczek, F., Hammond, P., and Cory, H. T. (1986), "The British Nationality Act as a Logic Program," *Communications of the ACM*, Volume 29, Issue 5, pp. 370-386.

Shanahan, M. (1997), *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*, MIT Press, Cambridge, MA.

Stranieri, A. and Zeleznikow, J. (1999), “The Evaluation of Legal Knowledge Based Systems,” *Proceedings of Seventh International Conference on Artificial Intelligence and Law*, Oslo, Norway, ACM Press, pp. 18-24.

Wahlgren, P. (2003), *Automation of Legal Reasoning*, Kluwer Law and Taxation Publishers, 1992.

Wang, J., *Distributed Information Organization and Management for Hazardous Waste Regulation Compliance Checking*, Ph.D. Thesis, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA..

Wos, L. and Pieper, G. (1999), *A Fascinating Country in the World of Computing*, World Scientific Publishing Co. Pte. Ltd..

Yabuki, N. (1992), *An Integrated Framework for Design Standards Processing*, Ph.D. Thesis, Department of Civil Engineering, Stanford University, Stanford, CA.

Zeleznikow, J. and Hunter, D. (1994), *Building Intelligent Legal Information Systems: Representation and Reasoning in Law*, Kluwer Law and Taxation Publishers.

Zohar, M. and Waldinger, R. (1993), *The Deductive Foundations of Computer Programming*, Addison-Wesley Publishing Company, Inc..