Automatic Rule-based Assessment of e-Government Application Forms

ANDY HON WAI CHUN
Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Kowloon Tong
HONG KONG
http://www.cs.cityu.edu.hk/~hwchun

Abstract: - This paper describes our current progress in designing and developing an AI rule-based module that automatically assesses e-Government application forms based on local laws and regulations. The rule-based module is part of a larger AI knowledge management (KM) system that is being designed to streamline the entire workflow processing for over a hundred different types of application forms that are handled by an Immigration agency. Each day, thousands of different types of application forms are submitted to this Government agency for processing, ranging from visas and identity cards to birth, death, and marriage certificate applications. Currently, this requires a substantially large workforce to manually process all the forms. This includes validating the data, collecting documents and then finally accessing whether the application can be approved or not. Our rule engine handles the difficult task of evaluating each application form to see if all legal regulations and guidelines have been met or not. This paper describes the design of this Assessment Rule Engine and how it works.

Key-Words: - Rule-based system, automatic assessment, expert system

1 Introduction

Immigration agencies play a very important role in maintaining the security and prosperity of a city. Firstly, it is responsible for controlling the entry and departure of all people at its borders and safeguarding it against threats. It is also responsible for enforcing immigration control within the city.

Besides immigration control, these agencies are also responsible for approving a wide variety of document applications ranging from right of adobe, travel documents, identity cards, to the registration of birth, death and marriage. For a city like Hong Kong, roughly 4 million of over a hundred different types of application forms were submitted for processing. This amounts to roughly 13,000 application forms per working day!

To handle these demanding workloads, the agency maintains a tight workforce of roughly 6,200 staff members. To overcome rapidly increasing workloads, it looks towards IT to improve efficiencies and productivities [1, 2, 3]. The automatic rule-based assessment of application forms is one such effort.

2 Application Processing Workflow

Since the agency handles over a hundred different types of application forms, the workflow for each will be slightly different. In general, in the current manual processing mode, hardcopy application forms are submitted together with photocopies of relevant documents. The applicant will be notified later to bring the original documents for verification. A suitable “case officer” will be assigned by the “authorization officer” to handle each application. After reviewing the application form, the case officer may request additional supplementary documents from the applicant. When all the supporting documents have been submitted, the case officer will then make an assessment for final endorsement by the authorization officer. The applicant will then be notified of the final result and return to collect permits or approval letters if application was successful.

In order for a case officer to adequate process an application, he/she must of course have thorough knowledge of all the applicable Hong Kong laws and regulations as well as immigration guidelines. In addition, the case officer must also be able to use his vast experience in processing other similar cases to draw on precedence cases for reference if discretionary decision-making [5] is needed.

Assessing a complex application can be very time consuming and knowledge intensive. The objective of our Assessment Rule Engine is to capture all the require knowledge of application assessment to help greatly reduce the time needed to process each form. Currently, depending on the application type, an application may require several days to several weeks to process. With the Rule Engine, applications may be processed in "one-stop" fashion or within a few days.

In the automated workflow, each application form and its associated supporting documents will either be submitted online or scanned and processed by optical character recognition (OCR), and then stored in a document management system (DMS). An initial preliminary assessment will then be made by our Assessment Rule Engine. For certain application types, all these tasks may be done while the applicant is still at the counter. For some application types, the process will be "one-stop" and the applicant can collect the permits/letters during the same visit.

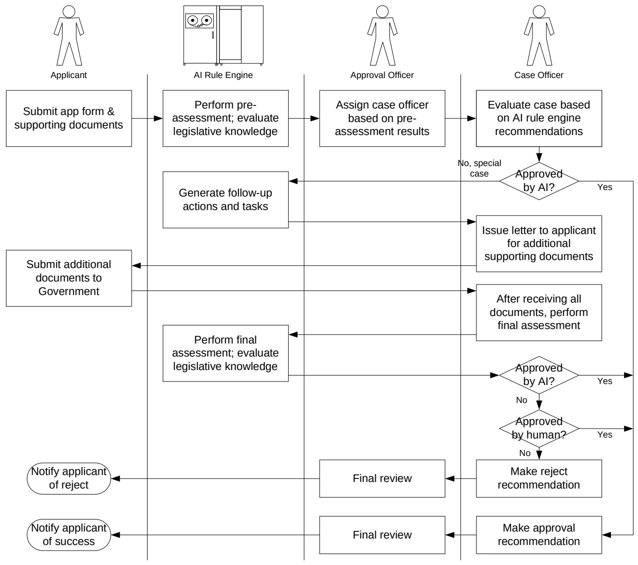
Applicant
AI Rule Engine
Approval Officer
Case Officer
Submit app form & supporting documents
Perform pre-assessment; evaluate legislative knowledge
Assign case officer based on pre-assessment results
Evaluate case based on AI rule engine recommendations
No, special case
Approved by AI?
Yes
Generate follow-up actions and tasks
Issue letter to applicant for additional supporting documents
Submit additional documents to Government
After receiving all documents, perform final assessment
Perform final assessment; evaluate legislative knowledge
Approved by AI?
Yes
No
Approved by human?
Yes
No
Notify applicant of reject
Final review
Make reject recommendation
Notify applicant of success
Final review
Make approval recommendation

Fig 1. The process flow with AI

For more complex applications, the preliminary assessment results will be used by the authorization officer to help him decide which case officer will be most suitable to assign to each case. Figure 1 shows the new process flow with AI. Once the case officer receives the application, he/she will use the Assessment Rule Engine to suggest some actions or steps to take. For example, the Rule Engine may recommend that the case officer request certain additional supplementary documents from the applicant. When all the documents are in place, the case officer will invoke the Assessment Rule Engine again to make a final assessment recommendation.

3 Related Work

The agency's use of an AI rule engine to perform automatic assessment is the first of its kind in the world. However, the use of rule engines or expert systems in other Government functions is not uncommon.

For example, the Australian Government [4] Information Management Office has been studying the use of rule engines to assist with decision making within the Australian Government. Its Business Process Transformation Committee (BPTC) oversees the reform of Australian Government business processes. The Automated Assistance in Administration Decision Making (AAADM) Working Group within the BPTC performed a study on the use of rule engines or expert systems in the Australian Government. It indicated the potential for cost savings, efficiencies and greater accuracy in decision making can be obtained with expert systems. It also indicated that the use of expert systems to assist administrative decision makers will become increasingly important.

Many agencies within the Australian Government are already using rule engines to assist with decision making. For example, the Department of Agriculture, Fisheries and Forestry uses rule-based systems to make decisions on whether to permit or reject an import, whether to perform import inspections, and what kind of tests to apply. The Australian Taxation Office also uses a large number of rule-based systems to assist in determining which methods should be used in calculating taxes, benefits, and penalties. Over 30 other rule-based applications are being developed. Customs uses expert systems to valuate importations, calculate customs tax, to profile and select high-risk import/export transactions for scrutiny. The Department of Defence uses rule-based systems to for workers compensation. The Department of Health and Ageing uses a rule-based system to check approved providers' compliances. The Department of Veterans' Affairs uses a rule-based system to support decision makers in determining veterans' entitlements.

In the United States, numerous expert systems are used to support the day-to-day operation of various government agencies. For example, the US

Customs and Border Protection agency uses an expert system called Automated Targeting System (ATS) [17, 18, 19] to find suspicious cargo transactions and for anti-terror work. ATS is a rule-based system with over 300 rules provided by field personnel, inspectors, and analysts in order to separate high-risk shipments from legitimate ones.

Our rule engine is similar to rules in traditional expert systems [6, 7]. However, instead of heuristics or rules of thumb, our rules encode legislative knowledge [8]. Each subsystem has its own rule base. The structure of our rule base was designed to facilitate easy of encoding expert knowledge on immigration-related legislations. A subsystem may have many different types of application forms. Each type of application has its rule agenda that defines which combination of rules or rule sets is applicable for a particular application type. Our rule agenda is similar to other rule agendas [9, 10, 11, 12] except that its main purpose is to encode relationships among rules rather than just sequence. Besides, rule agenda, rules are also organized into rule sets [10, 11, 12]. Each rule set represent one assessment criterion. Rules in the rule set represent how that criterion can be satisfied. Rules in our system operate in forward chaining manner [13].

4 Design of Assessment Engine

To streamline its application form processing workflow, the immigration agency uses a variety of AI technologies. This paper will only focus on the Assessment Rule Engine. This rule engine has the following key objectives:

- Perform initial preliminary assessment to assist workflow engine in routing application to the most appropriate case officer
- Perform continuous assessment to guide the process in collecting all necessary information and supplementary documents
- Perform final assessment to determine application result

All three tasks are processed using different features of the same rule engine and the same set of rules. The Assessment Rule Engine was designed to be used in the backend server to provide stateless rule-based processing services to the front-end Web clients. Therefore it has no GUI of its own; only a set of APIs.

4.1 Rule Engines

There is a separate Rule Engine for each "subsystem" within the Immigration agency. A subsystem represents a separate category of application types, for example, the "right of adobe" (ROA) application is one separate category or one subsystem. Within each subsystem, there may be many different specific types of application forms. For example, the ROA subsystem has 8 types of application forms.

Each subsystem rule engine is defined purely using RDF/XML documents [14]. The Java binaries are automatically generated directly from the RDF/XML documents using automatic source code generation technology. This greatly simplifies the maintenance of the rule engines. The RDF/XML documents can either be edited directly using a regular editor or a RDF/XML IDE tool, or via user friendly graphic user interfaces (GUI).

4.2 Rule Base

The rule base for each subsystem is defined using two RDF/XML documents – one to define all the rules and rule sets, and another to define the domain model. From these two RDF/XML files, a Java rule engine will be generated.

The rules are defined in terms of the following concepts:

- **Ruleset** – A ruleset is simply a collection of related rules. For the Immigration application, each ruleset represents a "criteria" to be satisfied before an application can be approved. For example, the criterion of whether an applicant is a Chinese citizen are not is represented by a set of rules that determine whether this criterion is satisfied or not. All rules related to determining Chinese citizen status will be in the same rule set.

- **Relations** – A relation represents a "fact" in the rule base. This fact can either be satisfied, failed, or undetermined. For example, a relation may be whether an applicant's name is the same on his/her travel document and the HK birth certificate. This relation will be true if the names are the same. It will be false if the names are different; and unknown status if data is not available to determine status. The ability to differentiate between these 3 states is an important feature for our rule engine; most software only differentiates between true/false or yes/no.

- **Rule** – A rule simply consists of a conjunctive and/or disjunctive set of relation statuses or rule statuses. The latter case allows us to define "meta-rules" – those are rules about rules. For the Immigration application, it represents the main rule governing the status of application. The individual rules within this main rule are the criteria rules. The criteria rules are in turn coded using relations.

- **Global Parameters** – Global parameters are used mainly in relations. It decouples the rule engine so that parameters need not be hard coded into the relations. Instead, they are stored separately in a global parameter object and can be changed by the user at any time to modify the behavior of the rule engine without changing the rules themselves. These are only used for parameters that changes frequently with time.

- **Domain Model** – The domain model represents the problem domain for each subsystem. This defines the types of information contained in each application form and the objects they are related to. For example, an application form may contain information related to the applicant, his/her parents, his/her employer, etc. The domain model also defines all the available attributes, the type of data that can be stored in the attributes, and the business language to be used to describe these attributes. The business language provides a vocabulary that can be used when defining the rules.

4.3 Rule Types

The Assessment Rule Engine was designed to encode rules related to laws, regulations, guidelines and actions. To support this, the system provides the following types of rules:

- **Hard Rule** – rule that indicates rigid assessment requirements that must be satisfied. These represent laws and regulations.

- **Soft Rule** – rule that has one or more conditions that are "soft"; can be violated but not desirable. These represent guidelines.

- **Action Rule** – rule that are used to suggest actions to be performed by the case officer. For example, request supplementary documents.

- **Verification Rule** – rule that require final verification by a human. For example, verifying information on a hardcopy document.

4.4 Rule Compilers

The rule compiler (rc) generates the Java binaries from the RDF/XML documents [14]; it automatically creates the application-specific rule engine Java jar file. There is no need to write or edit any Java source code.

Figure 2 illustrates how the rule engines are created. Rules and domain objects are defined using RDF/XML documents. These feed into our rule compiler that generates Java binaries. These binaries work on top of the CityU AI class libraries that contain classes to support AI and rule engine operations. The only Java coding that is needed is in creating code to call the auto-generated rule engine and to retrieve results for display to the user.

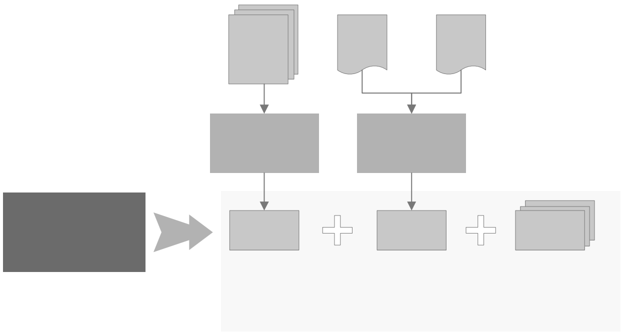

Fig 2. The Rule Compiler

5 Deployment Status

The design and development of system began early 2005 with the AI work starting mid-2005. The project prime contractor is NCSI, a wholly owned subsidiary of NCS, a leading IT solutions provider headquartered in Singapore with several thousand IT professionals worldwide. AI technology for the project was provided by the City University of Hong Kong.

AI deployment is prioritized based on subsystems and application types with the "Electronic Passport" [15, 16] and "Birth, Death, and Marriage" subsystems to be the first to be deployed.

The first version of the Assessment Rule Engine was released in mid-January 2006. Since then, the system has been undergoing extensive testing. At the same time, we have been customizing the AI engine for different subsystems and application types by setting configurations and encoding rules and parameters as well as fine-tuning features and performances.

User testing began in September 2006 with the first rollout to production in December 2006. Subsequent subsystems are scheduled to be deployed throughout 2007.

6 Summary and Conclusion

In this paper we presented the current progress in designing and developing a rule-based AI module to perform automatic assessment for e-Government immigration applications. The work documented here represents only part of the suite of AI technologies used in the assessment application.

References:

1. Enterprise Innovation, "The Hong Kong Immigration Department's continuing drive for automation", Aug 2006: http://www.enterpriseinnovation.net/perspectives.php?cat1=4&id=816

2. tdctrade.com, "HKSAR Immigration Department: aiming at security and quality service," May 2006: http://ict.tdctrade.com/suc-e513.htm

3. ComputerWorld Hong Kong, "Automation Fast-Track," July 2006: http://www.cw.com.hk/computerworldhk/article/articleDetail.jsp?id=352682

4. Administrative Review Council, "Automated Assistance in Administrative Decision Making," Report to the Attorney-General, Report No. 46, November 2004: http://www.ag.gov.au/agd/www/archome.nsf/Page/RWPD0BD2AED121A15A8CA256F5E00022956

5. Maria Jean J. Hall, Domenico Calabro, Tania Sourdin, Andrew Stranieri, and John Zeleznikow, "Supporting Discretionary Decision-Making with Information Technology: A Case Study in the Criminal Sentencing Jurisdiction," In the University of Ottawa Law & Technology Journal, Volume 2, Issue 1, 2005, pp.1-36.

6. Charles Forgy, John P. McDermott, "OPS, A Domain-Independent Production System Language," *Proc. IJCAI 1977*, pp. 933-939, 1977.

7. Bruce G. Buchanan and Edward H. Shortliffe, eds., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, MA, Addison-Wesley, 1984.

8. Anne von der Lieth Gardner, *An Artificial Intelligence Approach to Legal Reasoning*, Bradford Books/MIT Press, Cambridge, MA, 1987.

9. JBoss Rules, http://www.jboss.com/products/rules

10. Quintus Prolog, Quintus Flex, http://www.sics.se/isl/quintuswww/site/flex.html

11. Jess – the Rule Engine for Java Platform, http://herzberg.ca.sandia.gov/jess/

12. CLIPS – a Tool for Building Expert Systems, http://www.ghg.net/clips/CLIPS.html

13. Forward Chaining, Wikipedia entry, http://en.wikipedia.org/wiki/Forward_chaining

14. Resource Description Framework (RDF), W3C, http://www.w3.org/RDF/

15. Hong Kong to Issue e-Passport, Xinhua, 1 Jan 2007, http://english.gov.cn/2007-01/01/content_485516.htm

16. HKSAR Electronic Passport (e-Passport) and Electronic Document of Identity for Visa Purposes, http://www.immd.gov.hk/ehtml/eppt_edi.htm

17. Targeting Center the Brains behind Anti-Terror Efforts, US Customs and Border Protections, Nov/Dec 2005, http://www.cbp.gov/xp/CustomsToday/2005/nov_dec/targeting.xml

18. Border Security and Deterring Illegal Entry into the United States, April 1997, http://commdocs.house.gov/committees/judiciary/hju43664.000/hju43664_0f.htm

19. Homeland Security: Summary of Challenges Faced in Targeting Oceangoing Cargo Containers for Inspection, March 2004, GAO-04-557T, US Government Accountability Office, http://www.gao.gov/htext/d04557t.html