Using AI for e-Government Automatic Assessment of Immigration Application Forms

Andy Hon Wai CHUN

Associate Professor
City University of Hong Kong
Department of Computer Science
Tat Chee Avenue, Kowloon Tong
Hong Kong SAR
andy.chun@cityu.edu.hk

Abstract

This paper describes an e-Government AI project that provides a range of intelligent AI services to support automated assessment of various types of applications submitted to an immigration agency. The "AI Module" is integrated into the agency's next generation application form processing system which includes a workflow and document management system. AI services provided include rule-based assessment, workflow processing, schema-based suggestions, data mining, case-based reasoning, and machine learning. The objective is to use AI to provide faster and higher quality service to millions of citizens and visitors in processing their requests. The AI Module streamlines processes and workflows while at the same time ensuring all applications are processed fairly and accurately and that all relevant laws and regulations have been considered. It greatly shortens turnaround time and indirectly helps facilitate economic growth of the city. This is probably the first time any immigration agency in the world is using AI for automatic application assessment in such a large and broad scale.

Task Description

Immigration agencies play very important role in maintaining the security and prosperity of a city. They control the entry and departure of people at its borders and safeguard it against threats. They may also be responsible for enforcing immigration control within the city.

Besides immigration control, the immigration agency that we performed this work for is 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. In 2004, close to 4 million of over a hundred different types of application forms were submitted to the agency for processing.

The agency currently maintains a tight workforce of roughly 6,200 staff members. To overcome rapidly increasing workloads, it looks towards IT to improve efficiencies and productivities [2, 4]. The agency has successfully launched numerous award-winning IT projects to streamline immigration operations as well as further improve the quality of service provided to its citizens. Notable projects include, for example, using smart identity cards and biometrics verification to get citizens through its borders quickly, an entry/exit processing system that facilitates speedy clearance of passengers at control points, a face recognition system that helps identify persons attempting to use different identities at immigration control, a system to expedite processing of passports and other travel documents, and many more.

Our project is part of a new IT strategy that includes roughly 30 projects total. The application form processing system and associated document management system, which our AI work [1, 3] is part of, represent the 3rd phase of this strategy, and cost over US$40 million to develop [7]. The systems provide the agency with a paperless environment, streamlined with AI, to cope with continued growth and increasing workloads.

With the new systems, the public will be able to submit most applications electronically. Visits to the agency will be minimized and processing time will be significantly shortened. One-stop on-the-spot service will be provided for some types of application. Progress of applications can of course be checked through the Web anytime. This greatly improves the level of convenience to citizens.

The software is divided into several "subsystems" representing different classes of applications, for example, the "Right of Adobe", "Certificate of Entitlement", "Birth, Death, and Marriage" (which includes adoption), "Permits and Visas", "Travel Pass System", "Investigation", "Nationality", "Assistance to Residents", and "Electronic Passport" subsystems. Each subsystem has its own customized version of our AI Module.

Current Manual Approach

The workflow for each type of application is slightly different. Here, we will describe a typical workflow for the manual approach. The process starts with the applicant appearing in person to submit a hardcopy application form together with photocopies of relevant documents; the original documents will be verified later. The "authorization officer" reviews the case and assigns a suitable "case officer," who is familiar with handling that type of application, to be responsible for that case. After reviewing the application form, the case officer may request additional supplementary documents from the applicant. When all the supporting documents have been submitted and verified, the case officer will make a final assessment, which will then be reviewed and endorsed by the authorization officer. Finally, the applicant will be notified of the result and return to collect permits or approval letters if application was indeed successful. The entire process may require several visits by the applicant to the immigration agency and many days, weeks or even months to complete depending on complexity.

In order for a case officer to adequately process an application, he/she must of course possess thorough knowledge of all the applicable 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 is needed. Assessing a complex application can be very time consuming and knowledge intensive. The objective of our AI work is to capture the different types of knowledge required for application assessment so that they can easily be used by the case officer to quickly formulate an informed and accurate decision.

New AI Approach

With the new AI system, application forms will either be submitted online or as hardcopy and then scanned and processed by optical character recognition. Associated supporting documents will also be stored digitally in a secured document management system. Online submission represents substantial savings in community cost since the applicant need not go to the agency in person [7].

After submission, an initial preliminary assessment will be made by the AI Module's Assessment Rule Engine that contains a knowledge-base of all applicable laws and regulations as well as immigration guidelines for each application type.

For certain types of application, the assessment may be done in a "one-stop" fashion and the applicant can collect the permits/letters during the same visit. For more complex applications, the preliminary assessment results from the AI Module will be used by the authorization officer to decide which case officer will be most suitable to handle that case. Once the case is passed to the case officer via the workflow system, he/she will use the AI Module's Schema-based Reasoning Engine to generate a set of suggested actions or steps to take in order to get the application to a final state that can be assessed. For example, the AI Module may recommend that the case officer request certain additional supplementary documents from the applicant. The schema encodes procedural knowledge of typical steps or actions taken by case officers in handling different kinds of cases.

The workflow for each application is managed using the AI Module's Workflow Rule Engine that determines how each case should be routed through the immigration processes. For special or non-typical cases, the AI Module provides a Case-based Reasoning (CBR) Engine [38] to analyze and retrieve similar precedence cases for the case officer to review and compare. When all the necessary information and documents are finally in place, the case officer can invoke the Assessment Rule Engine once again to make a final assessment recommendation. Since discretionary decision making is used for complex cases, the AI Module contains a Self-Learning Engine that performs learning on decisions made and then reuses this newly learned knowledge (in the form of "self-learned rules") when processing new applications. Once applications are processed and final outcome decisions have been made, they will be indexed into our AI case base using AI clustering [14].

Related Work

The immigration AI work is built upon several different AI representations and reasoning algorithms - rules, schema-based, clustering, case-based, and decision trees.

Our rule engine is similar to rules in traditional expert systems [16, 17]. However, instead of heuristics or rules of thumb, our rules encode legislative knowledge [18]. 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 [19, 20, 21, 22] 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 [20, 21, 22]. Each rule set represent one assessment criterion. Rules in a rule set represent how that criterion can be satisfied. Rules in our system operate in forward chaining manner [23].

Many government agencies around the world use rule engines to assist with decision making, such as the Australian Government [30]. 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 number of rule-based systems to assist in determining which methods should be used in calculating taxes, benefits, and penalties. 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 calculate 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 US, the Customs and Border Protection agency uses an expert system called Automated Targeting System (ATS) [27, 28, 29] to find suspicious cargo transactions and for anti-terror work. ATS has over 300 rules provided by field personnel, inspectors, and analysts in order to separate high-risk shipments from legitimate ones.

Besides rules, our AI Module uses schema-based reasoning [13, 31] to represent procedural knowledge of actions and tasks that the case officers may take in the course of handling a case. For example, request for verification of certain documents, letters of reference, etc. Actions and tasks are triggered by rules.

Schema-based reasoning was also used in SAIRE [32], a multi-agent AI search engine to search Earth and space science data over the Internet. Chen and Lee [33] explored how schema-based reasoning can identify fraud potentials exposed by an internal accounting control system.

To provide decision support and precedence case retrieval, our AI Module make use of incremental [15] AI clustering [14] with multi-valued attributes [34] using k-means clustering algorithm [35]. AI clustering has been used successfully for many similar applications, such as QCS [36] – an information retrieval system that allows users to retrieve relevant documents separated into topic clusters with a single summary for each cluster. IBM Research [37] developed a clustering system for indexing, analysis, and retrieval of videos.

Our case-based reasoning [38] engine makes use of AI clustering results to retrieve similar relevant cases to create a recommendation and summaries. CBR is a popular approach to reuse previous experience to handle new situations. For example, PlayMaker [39] is a CBR prototype that models how air traffic controllers handles traffic flow under severe weather or congestion. Xu and Li [42] used CBR to identify people who are "AIDS risky" and could be target for intervention. Esmaili and others [43] used CBR for intrusion detection.

Finally, we use incremental decision trees [40, 41, 44] to perform machine learning and rule generation [45] to capture how case officers handle "difficult" cases. In Australia, the Department of Family and Community Services has an "Edge expert system" that uses decision trees to determine a citizen's likely entitlement to payments and services [30].

Application Description

The platform for the application processing system is Java EE. The AI Module is therefore also Java-based and packaged/deployed as Java EAR files. For scalability, AI services are provided in a stateless manner to the application processing system and can be deployed on as many application servers as needed.

The front-end to our AI software is a Web-based thin client operated by immigration case officers. The layout and design of the Web client is typical of other form-based systems. Each case officer has an inbox containing all the applications he/she has been assigned to handle. For each application, there are several screens to display personal information on the applicant, the details of the current application, related documents provided by applicant and those sent to applicant by the immigration agency, historical record on this applicant such as other applications he/she has submitted before, other related information and follow up actions. Basically, anything related to an applicant, all his current and past applications and documents are all consolidated in a conveniently accessible dashboard for the case officer to review.

The various AI features used by case officers operate in near real-time within a few seconds or so. The other AI tasks that are not performance critical, such as case learning and rule generation are done behind the scene as background processes.

AI processing results are displayed on two key screens – Assessment Screen and Decision Support Screen. The "Assessment Screen" displays results of the Assessment Rule Engine as a list of violated rules and details of those rules, such as attributes and parameter values as well as links to legal references related to that rule. Rules may include hard rules, soft rules, or self-learned rules. It also contains a set of recommended follow up action to take (generated by the Schema-based Reasoning Engine), such as requesting additional documents or verifying validity of certain information provided by the applicant. Actions already taken are also displayed.

The "Decision Support Screen" is used to handle "special" or non-typical cases. It displays a list of related precedence cases and their key attributes, generated by the Case-based Reasoning Engine through AI clustering. In addition, the case officer can request the CBR Engine to search for similar cases based on a selected subset of those attributes.

Once all the necessary actions and steps have been taken and all the relevant documents have been collected, the AI Module will be used again to generate a final assessment.

AI Architecture

Figure 1 shows a high-level view of our AI architecture. It consists of several key technologies and modules:

- Rule Engines:
  - Assessment Rule Engine
  - Workflow Rule Engine
  - Schema-based Reasoning Engine
- Case-based Reasoning Engine
- Self-learning Engine
- Knowledge Base
  - rules, schemas, cases and decision trees

Each and every application form submitted to the immigration agency will be processed by the AI Module. The results from AI processing may include pre-assessment results, follow-up actions, links to legal references, results from self-learned rules, precedence cases, workflow routings and final recommendations.

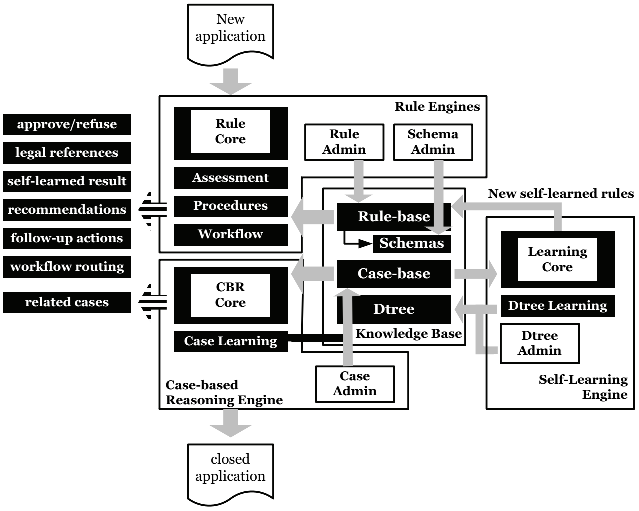

Figure 1. Overall AI architecture.

This architecture is replicated for each subsystem within the application processing system. For example, the electronic passport subsystem has its own set of rule, case-based, and self-learning engines as well as knowledge base and databases. This makes deployment easier as well as improves maintainability and scalability.

Automatic Code Generation

One of our design objectives is easy of maintenance – to ensure that rules can be updated easily and that changes in the rule engine will not affect other components. To achieve this, we took a “non-intrusive” approach to AI where the customized AI engines for each subsystem are defined using only RDF/XML documents [8]. Automatic code-generation techniques were then used to dynamically generate the AI engines as Java binaries and associated database tables. The generated AI engines are decoupled from other components. This further shortens development time and minimizes potential coding errors [9]. Many popular rule engines already use XML to encode rules [10, 11, 12]. We use this approach extensively for all the AI engines, not just the rule engines. Furthermore, the interface codes to the relevant domain objects that represent the application details are also auto-generated from RDF/XML and require no Java coding. In the past, interfacing an AI engine to an application is usually very time consuming and error prone. To further simplify interfacing, results from AI processing are simply returned as an encapsulated result object.

Figure 2 illustrates the structure of the AI Module for each subsystem. In the center is the “AI Core” – a collection of AI classes representing engines, algorithms and routines developed at CityU. Wrapping around the AI Core are subsystem-specific AI engines that are created using our “engine compilers” that converts RDF/XML to Java jar files and databases. The only Java coding that is needed is in the “Shared AI Coding” – a common set of features used by all subsystems, and the “Subsystem Custom Coding” – behaviors and flows specific to a particular subsystem.

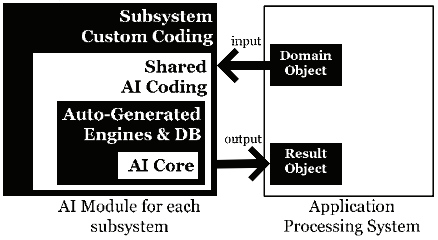
Subsystem Custom Coding
Shared AI Coding
Auto-Generated Engines & DB
AI Core
input
output
Domain Object
Result Object
AI Module for each subsystem
Application Processing System

Figure 2. AI Module structure for each subsystem.

Although there are many subsystems and over a hundred types of application forms, our development approach makes the AI Module very easy to customize and maintain. Figure 3 highlights how the key AI software components are created. The main work is performed by the AI “engine compilers” that automatically generates the various AI engines, object/relational mapping and associated databases from RDF/XML without coding. The AI Core (basic AI features) and Shared AI Coding (immigration-related features) are shared among all subsystems and requires no coding for deployment to a new subsystem. The only coding is in the Subsystem Custom Coding that encodes subsystem-specific processing and flow. The size of this coding is relatively small.

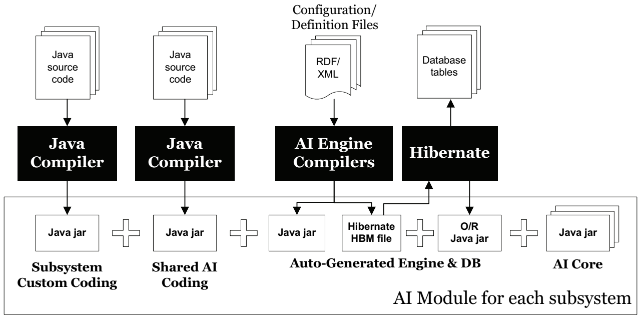

Figure 3. Software components in each subsystem's AI Module.

Uses of AI Technology

In order to achieve all the objectives of the project, several different AI techniques were used.

The Assessment Rule Engine

The Assessment Rule Engine is probably the most important as it encodes immigration-related legislative knowledge and guarantees that all applicable laws, regulations and guidelines have indeed been considered for each and every case. The key functions are:

- Perform initial preliminary assessment to assist workflow engine in case assignment as well as guide information collection
- Perform final assessment to determine application result

There is one engine and one rule base per subsystem. Each subsystem may have many different types of application forms. In total, there are several thousand rules in the AI Module to handle all the subsystems.

To organize this large body of legislative knowledge, the rule base for each subsystem contains a separate "rule agenda" for each application type. The agenda determines which combination of rule sets is applicable for a particular application type. Rule sets contain all the rules related to determining the status of particular "assessment criteria." For example, whether a person is a recognized "citizen" or not is one criterion in determining his/her right of adobe. For this criterion, there are over thirty different rules to help determine whether that criterion is satisfied or not. All those rules are stored in the "citizen" rule set and controlled by agendas that require this criterion in assessment of their associated application type.

Although most laws and regulations regarding immigration are relatively static, some of the guidelines do change from time to time. To facilitate user maintenance of rules without having to regenerate and republish a new rule engine each time, the rules are designed with parameter-driven capabilities. Parameter values can be edited by users with appropriate authority; the effect on related rules is instantaneous.

Besides established legislative knowledge, the rule engine also uses knowledge on discretionary decision-making. This knowledge is taken from "self-learned" rules that are generated by the Self-Learning Engine from observing how case officers handle "special" cases.

Evaluation: Before AI, assessments were done by sorting through and reviewing paper documents submitted by the applicant and using the case officer's own personal experience and knowledge of laws and guidelines. Time needed might be a few minutes for very simple cases to much longer for complicated cases. Some cases may take several days as the case officer may need to seek help and advice from other officers or superiors. With AI, assessment is done in less than 10 seconds for all cases regardless of complexity, while guaranteeing that all relevant legislations and guidelines, as well as all possible approval scenarios are considered.

The Schema-based Reasoning Engine

The schemas stored in the Schema-based Reasoning Engine [13] represent procedural knowledge in processing applications. It is used to:

- generate tasks, checklists and follow up actions for the case officer to perform.

It guides the case officer in collecting of all necessary information and supplementary documents as well as printing documents and instructions. The engine is itself rule-driven. This allows different sets of steps and actions to be proposed depending on the particulars of the application at hand and previous actions already taken.

Evaluation: Before AI, if there are unclear points in the application or if certain information needs to be verified, the case officer sends a letter to the applicant for additional supporting documents. After receiving the documents, the case officer analyzes the case once again and possibly requests more information from the applicant if needed. This cycle is time-consuming and stressful for the applicant as he/she may need to visit the agency several times before his/her application can be assessed. With AI, different scenarios are analyzed automatically at the same time and a consolidated list is generated, thus minimizing the number of visits an applicant needs to make to the agency. Furthermore, letters to applicants are generated automatically. A task checklist is also provided to keep track of tasks so that nothing is overlooked.

The Case-based Reasoning Engine

Straightforward cases are handled automatically by the Assessment Rule Engine. But in the real world, there are many "special" cases that require more detailed analysis. This complicates and lengthens the assessment process. The CBR Engine helps alleviate this situation by:

- retrieving relevant "closed" cases from the case-base to act as precedence or reference
- indexing newly closed cases into the case-base

There is one CBR Engine per application type since the attributes considered by different application types will be different. Each case is represented by a prioritized attribute vector that contains either data from the application form or results from the Assessment Rule Engine. The objective of the CBR is to retrieve similar cases and use statistics to generate recommendations. The CBR Engine also supports advanced features, such as multi-valued attributes and incremental AI clustering [15].

The case officer may fine-tune the way the CBR Engine retrieves relevant cases by selecting the assessment criteria that he/she feels are most important for the case at hand.

Evaluation: Before AI, it was very difficult if not impossible to locate past cases to use as reference or establish precedence. There was no easy way to search through case folders, which were stored in physical archives. Case officers had to rely on their own memory or personal notes, or ask other case officers if they remember handling similar cases before. Even if cases can be recalled, trying to retrieve it from archives takes time. With AI, a precise list of all similar cases within a given time period can be retrieved within seconds, with all the details of the cases, analysis results and comments from case officers in charge.

The Self-Learning Engine

The Self-Learning Engine captures discretionary decision-making knowledge that represents case officers' experience in handling special cases as well as their knowledge of assessment best practices and guidelines that change with the changing needs of our society. The key functions are:

- incrementally learn and index new cases into decision trees
- generate self-learned rules from the decision trees and integrate into the Assessment Rule Engine

For the same reasons as the CBR Engine, there is one Self-Learning Engine per application type. However, each engine may contain many decision trees. Each decision tree represents knowledge related to one assessment criteria. These are the same assessment criteria used by the rule engines as well as the CBR engines. The decision tree is constructed from prioritized data either from the application form or retrieved from results from the Assessment Rule Engine. This engine also supports advanced features such as incremental learning.

The self-learned rules generated from each decision tree are used to determine whether an assessment criterion was fulfilled or not. Hence rules generated by the Self-Learning Engine are well integrated with the Assessment Rule Engine and directly contribute to the assessment result.

Evaluation: Before AI, discretionary decision-making was based on best practices and guidelines that were discussed and shared informally among case officers. Each case officer keeps a personal notebook of these guidelines and practices as reference. The performance of a case officer will depend greatly on his/her knowledge of these best practices and guidelines and his/her personal experience in handling similar cases. There was no easy way to share this type of knowledge efficiently before. With AI learning, patterns in discretionary decision-making are extracted and codified as rules so that best practices can be shared and used regardless of the experiences of the case officers.

Application Use and Payoff

The AI Module was deployed to production in December 2006. Starting early February 2007 it began to process each and every application for electronic passport. Rollout for remaining subsystems is scheduled throughout 2007. So far, several hundred immigration case officers have been trained on the system.

Evaluation Results

Prior to deployment, extensive unit, integration, and stress testing were performed. After that, the system went through 2 months of user testing and 2 months in production environment before official launch in February 2007. Obviously, for this type of AI system, it must return correct results 100% of the time and be fast (within seconds) and stable. The following summarizes general feedbacks and results from user evaluation:

- For subsystems with large volume of applications, automatic assessment with AI rules is the only way to improve efficiency
- The ability to automatically find precedent cases is very important and highly useful for decision support, which was practically impossible to do manually.
- Self-learning is also very important for certain application types because rules can be too complex to code manually.
- Automatically consolidate all information related to a particular case into a dashboard was found to be very useful. The old approach of manually sorting through paper documents and records was too time consuming and error-prone.
- The ability to automatically propose follow-up actions and to automatically generate notification letters and minutes was also found to be very useful and a major time savings.

Key payoffs include:

- **Quality of Service** – This is number one priority for this immigration agency. Year after year, it receives numerous awards and recognitions for outstanding quality of service to citizens and visitors [5]. The use of AI to streamline processing workflow enhances the quality of service by reducing turnaround time [6]. For example, time to get an entry permit for employment can be shortened by 3 to 5 days whereas a search of births, deaths or marriage records can be reduced to several minutes. One-stop service is now possible for most applications. Secondly, the use of AI provides a more comprehensive and thorough assessment of each case so that follow up tasks are consolidated. Therefore the number of documents that the applicant must provide and the number of visits to the agency is minimized.
- **Quality of Assessment** – In the past, assessment quality depended on the experience and knowledge of case officers. Time was needed to think through numerous intricate and complex laws and regulations for each type of application. Manual assessment was time-consuming and error prone. With AI, all relevant laws, regulations and guidelines are considered at all times within seconds, guaranteeing that nothing is overlooked and eliminates any potential for errors.
- **Increased Productivity –** For complicated cases, case officers need time to sort out all the information provided by the applicant as well as run through different approval scenarios. This can be time-consuming and may require discussions with other case officers to clarify fine-details of legislation. With AI, applications are assessed under all possible scenarios at the same time and within seconds. In addition, retrieving historical case folders from physical archives was very hard to do before. Using AI, relevant cases are automatically retrieved without any effort from the case officer. Case officers can then focus on using his expertise more effectively for decision making.
- **Improved Agility** – Because the AI Module is parameter-driven, any urgent change in guidelines and policies can be made instantly without any change to software. With self-learning capabilities, the AI Module automatically adapts itself to current practices and guidelines. Hence the agency becomes

more agile in terms of its knowledge management capabilities.

- **Capacity for Growth** - In the long term, the AI Module will allow the agency to cope with continuously increasing workloads to support the city's economic growth.
- **Economic Savings** – The agency estimates that the application processing system will save the Government over US$16 million annually [7]. Efficiencies provided by the AI Module not only represent cost savings for the Government but also substantial savings in community cost in reduced waiting and turnaround time for its citizens and visitors.

Application Development and Deployment

The design and development of the application processing 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.

The total IT team size for the entire project consists of roughly 200 programmers, system analysts and consultants from several IT vendors and system integrators around the world. In addition, roughly another 60 officers and managers from the user side are dedicated to this project.

The AI design and development team consists of roughly ten knowledge engineers and AI developers. AI development was simplified with extensive support from the user side in providing knowledge in a form that was readily convertible into rules for the rule engine.

For a system as complex as this, integration, robustness and scalability were major concerns when we designed the AI Module. To minimize integration issues, the AI Module was designed to be decoupled from other components using well-defined interfaces. Robustness is handled by designing the AI Module to be deployable per subsystem or even per application type if needed. Thus any fault in one subsystem or application type will not affect others. In addition, all internal databases used by the AI Module have redundancy to improve robustness and performance. Scalability is handled by designing the AI Module to provide AI services in a stateless manner. If workload increases, all that is needed is simply to add more application servers. This distributed design also allows the application to switch over to another AI server if one fails.

Deployment

AI deployment is prioritized based on subsystems and application types with the "Electronic Passport" [25, 26] and "Birth, Death, and Marriage" subsystems to be the first to be deployed.

The first version of the Assessment Rule Engine and Schema-based Reasoning Engine was released in mid-January 2006. This was followed by the CBR Engine in mid-February 2006 and the Self-Learning Engine in end-March 2006. Since then, the systems have been undergoing extensive testing. In parallel, the engines were customized for different subsystems and application types 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.

Maintenance

Just like any other mission critical software, there will inevitable be changes and upgrades to the AI Module after deployment to reflect legislative and/or operational changes for the agency. The design of the AI architecture is such that these types of changes are very easy to do.

Firstly, all knowledge-related changes can be done without any Java coding and simply by updating RDF/XML documents and configuration files. Binaries and databases are generated automatically by our "engine compilers." Secondly, the behavior of the rule engines are parameter-driven and under user control to reduce the need for code change. Packaging the AI Module as a decoupled component from the other parts of the system helps further reduce maintenance and integration needs.

For support, the prime contractor NCSI's IT team provides front-line technical and end-user support while the CityU provides additional assistance on the AI technologies when needed.

Conclusion

This paper provides an overview of how various AI techniques were used to provide highly intelligent and accurate case assessment capabilities to an e-Government system for automated processing of immigration-related application forms. AI streamlines processes and results in higher quality and faster service to citizens and visitors. In addition, valuable domain knowledge and expertise related to immigration laws, regulations and guidelines are now quantified, coded and preserved within the agency, for use in this and other systems. Our AI work makes use of several innovative techniques, such as non-intrusive RDF/XML coding and integrated rule, schema-based, case-based, and self-learning engines as well as incremental clustering and learning. This is mostly likely the first time any immigration agency in the world is using AI for automated intelligent application assessment and in such a large deployment scale.

References

1. Enterprise Innovation, "The HK 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 HK, “Automation Fast-Track,” July 2006:http://www.cw.com.hk/computerworldhk/article/articleDetail.jsp?id=352682
4. Immigration Department Annual Report 2004-2005: http://www.immd.gov.hk/a_report_04-05/htm_en/ch05/url/frame_020.htm
5. Immigration Dept of HK (website), “Recognition of the Quality Service of Immigration Dept”: http://www.immd.gov.hk/ehtml/rqs.htm
6. Immigration Dept of HK (website), “Performance Pledge”:http://www.immd.gov.hk/ehtml/pledge.htm
7. Implementation of Phase III of the Updated Information Systems Strategy for the Immigration Dept:http://www.legco.gov.hk/yr03-04/english/fc/fc/papers/f04-10e.pdf
8. RDF/XML Syntax Specification (Revised), W3C Recommendation 10 Feb 2004: http://www.w3.org/TR/rdf-syntax-grammar/
9. Said Tabet, Prabhakar Bhogaraju, David Ash, “Using XML as a Language Interface for AI Applications,” PRICAI Workshops 2000, Springer-Verlag, 2000, pp.103-110.
10. A Java Deductive Reasoning Engine for the Web (jDrew) (n.d.), http://www.jdrew.org/
11. The Mandarax Project (n.d.), http://mandarax.sourceforge.net/
12. Drools: Object-Oriented Rule Engine for Java (23 January 2004), codehaus, http://drools.org/
13. “What are Schemas?” in Elise H. Turner, *A Schema-based Approach to Cooperative Problem Solving with Autonomous Underwater Vehicles,* http://bronte.umcs.maine.edu/Papers/1991/oceans/subsectionstar3_1_1.html
14. Fisher, D., Xu, L., Carnes, J., Reich, Y., Fenves, S., Chen, J., Shiavi, R., Biswas, G., & Weinberg, J. “Applying AI Clustering to Engineering Tasks,” *IEEE Expert*, 8, pp.51-60, 1993.
15. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Michael Wimmer, Xiaowei Xu, Incremental Clustering for Mining in a Data Warehousing Environment, *Proc. 24th Int. Conf. Very Large Data Bases*, VLDB, 1998.
16. Charles Forgy, John P. McDermott, “OPS, A Domain-Independent Production System Language,” *Proc. IJCAI 1977,* pp. 933-939, 1977.
17. 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.
18. Anne von der Lieth Gardner, *An Artificial Intelligence Approach to Legal Reasoning*, Bradford Books/MIT Press, Cambridge, MA, 1987.
19. JBoss Rules, http://www.jboss.com/products/rules
20. Quintus Prolog, Quintus Flex, http://www.sics.se/isl/quintuswww/site/flex.html
21. Jess – the Rule Engine for Java Platform, http://herzberg.ca.sandia.gov/jess/
22. CLIPS – a Tool for Building Expert Systems, http://www.ghg.net/clips/CLIPS.html
23. Forward Chaining, Wikipedia entry, http://en.wikipedia.org/wiki/Forward_chaining
24. Resource Description Framework (RDF), W3C, http://www.w3.org/RDF/
25. Hong Kong to Issue e-Passport, Xinhua, 1 Jan 2007, http://english.gov.cn/2007-01/01/content_485516.htm
26. HKSAR Electronic Passport (e-Passport) and Electronic Document of Identity for Visa Purposes, http://www.immd.gov.hk/ehtml/eppt_edi.htm
27. 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
28. Border Security and Deterring Illegal Entry into the United States, April 1997,http://commdocs.house.gov/committees/judiciary/hju43664.000/hju43664_0f.htm
29. 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
30. 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
31. Turner, R. M., *Aaptive Reasoning for Real-World Problems: A Schema-Based Approach*, Lawrence Erlbaum Assoc, Hillsdale, NJ, 1994.
32. Odubiyi, J.B., Kocur, D.J., Weinstein, S.M., Wakim, N, Srivastava, S., Gokey, C., & Graham, J, “SAIRE- A Scalable Agent-Based Information Retrieval Engine,” In *Proc of the 1st Int’l Conf on Autonomous Agents,* Feb 5-8, 1997, pp.292-299.
33. Chen, K.T.; Lee, R.M., *Schematic evaluation of internal accounting control systems,* EURIDIS Research Monograph, RM-1992-08-1. Erasmus University, Rotterdam, August 1992
34. T.W. Ryu, C.F. Eick, “Similarity measures for multi-valued attributes for database clustering,” In *Proc of the Conf on Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Rough Sets*, St. Louis, MO, 1998.
35. K-Means Clustering Algorithm, Wolfram MathWorld, http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html
36. Daniel M. Dunlavy, Dianne P. O'Leary, John M. Conroy, and Judith D. Schlesinger, *QCS: A System for Querying, Clustering, and Summarizing Documents*, SANDIA Technical Report, July 2006.
37. Murray Campbell, Alexander Haubold, Shahram Ebadollahi, Milind R. Naphade, Apostol Natsev, Joachim Seidl, John R. Smith, Jelena Tešić, and Lexing Xie, "IBM Research TRECVID-2006 Video Retrieval System," In NIST TRECVID-2006 Workshop, Gaithersburg, MD, Nov 2006.
38. Kolodner, Janet, *Case-Based Reasoning,* San Mateo, Morgan Kaufmann, 1993.
39. Kenneth R. Allendoerfer, Rosina Weber, *PlayMaker: An Appl. of Case-Based Reasoning to Air Traffic Control Plays*, ECCBR, 2004, pp.476-488
40. T. Mitchell, "Decision Tree Learning", in T. Mitchell, *Machine Learning*, McGraw-Hill, 1997, pp. 52-78.
41. P. Winston, "Learning by Building Identification Trees", in P. Winston, *Artificial Intelligence*, Addison-Wesley, 1992, pp. 423-442.
42. Xu L, Li N., “Applying rule- and case-based reasoning to AIDS initial assessment,” In *Proc of the Int Conf AIDS,* Jul 7-12, 1996; 11: 475.
43. Esmaili, M. Balachandran, B. Safavi-Naini, R. Pieprzyk, J., “Case-based reasoning for intrusion detection,” In *Proc of the Computer Security Applications Conference*, San Diego, CA, USA, 9-13 Dec 1996, pp.214-223.
44. Utgoff, P.E., “Incremental induction of decision tree” *Machine Learning*, 4:161-186, 1989.
45. Quinlan, J.R., “Generating Production Rules from Decision Trees,” In the *Proc of IJCAI 1987,* pp.304-307.