An Ontological Inference Driven Interactive Voice Recognition System

Mohammad Ababneh
Department of Computer Science
George Mason University
Fairfax, VA, USA
mababneh@gmu.edu

Duminda Wijesekera
Department of Computer Science
George Mason University
Fairfax, VA, USA
dwijesek@gmu.edu

***Abstract*— Someone seeking entry to an access controlled facility or through a border control point may face an in person interview. Questions that may be asked in such an interview may depend on the context and vary in detail. One of the issues that interviewers face is to ask relevant questions that would enable them to either accept or reject entrance. Repeating questions asked at entry point interviews may render them useless because most interviewees may come prepared to answer common questions. As a solution, we present an interactive voice response system that can generate a random set of questions that are contextually relevant, of the appropriate level of difficulty and not repeated in successive question answer sessions. Furthermore our system will have the ability to limit the number of questions based on the available time, degree of difficulty of generated questions or the desired subject concentration. Our solution uses Item Response Theory to select questions from a large item bank generated by inferences over multiple distributed ontologies.**

Keywords—Ontology; Semantic Web; OWL; Dialogue; Question Answering; Voice Recognition; IVR; VXML; Access Control Policy; Security; Item Response Theory.

I. INTRODUCTION

Physical control points such as human guarded gates, border control points and visa counters provide entry into facilities or geographical regions to those that can be *admitted legitimately*. Legitimacy is usually determined by rules, regulations or policies known to entry control personnel whose duty is to ensure that these policies are enforced while admitting people. In order to do so, they hold an interview, in which an aspiring entrant is asked a series of questions, and possibly show some documents and demonstrate some knowledge about the contents of the documents or attributes contained in them. Successful interviews should have questions that are relevant, of a reasonable level of difficulty (i.e. not too difficult or common knowledge) and not to have been asked in prior interviews for the same purpose without drawing accusations of bias from rejected entrants. Ideally, a successful interview should accommodate differences in accents and provide assurance that it is unbiased against similar attributes.

Given the recent success of interactive voice response (IVR) systems such as auto attendants, satellite navigation, and personal assistants such as Apple's Siri, Google's Voice, Microsoft's Speech, we investigated the possibility of specializing IVR systems for access control such as: Visa interviews, entry point interviews, biometric enrollment interviews, password reset, etc.

Although IVR systems have come a long way in recognizing human voice, and responding to human requests as if responses come from another human, most of the existing IVR systems are pre-programmed with questions and their acceptable answers, and consequently have limited capability in satisfying the Use Case at hand.

The first minor limitation of current IVR systems comes from the fact that, the human starts and drives the conversation. The second limitation is that most IVR systems have a finite number of pre-programmed conversations. Therefore the set of questions generated by such a system are the same for every conversation. This limitation may expose the set of questions so that aspiring entrants may come with prepared question-answer pairs, even if the subject matter of the questions may be unfamiliar to them. Consequently, having the ability to select questions from a large pool may resolve this limitation. The third limitation is that when selecting a random set of questions from large pool is that the set of questions asked may not have the desired overall level of difficulty to challenge the user. Solving this issue is relevant because all aspiring entrants expect to have a fair interview. The forth limitation is that questions must be able to discriminate between someone that knows the subject matter from someone who guesses an answer.

As a solution we created an ontological inference based IVR system that uses item response theory (IRT) to select the questions [13, 3]. Our system uses the XACML language as a base to establish entry policies that consist of rules to specify the attributes that must be possessed by permitted entrants [7]. The IVR system has the responsibility of determining access by asking questions generated using ontological inferences and IRT.

In previous work, we introduced a policy-based IVR system for use in access control to resources [1]. Later, we presented an enhancement that uses IRT to select queries from a large set of attributes present in a policy [2]. Here we introduce ontology-aided access control system by including questions related to the base attributes in order to ascertain the interviewee's familiarity, and provide a score for the entire set of answers [8]. We also have the added capability to generate the succeeding question based on the accuracy of the preceding question. We do so by aligning each attribute with an ontology that encodes the subject matter expertise on that attribute and derive facts from these ontologies using reasoners to generate questions. We then assign weights to these derivations based on the axioms and rules of derivations used in the proof tree.

Usually ontologies have a large number of axioms and assert even more facts when using reasoners. Consequently, blindly converting such an axiom base to human-machine dialogue would result in very long conversations with many disadvantages. The first is that human users would become frustrated of being subjected to long machine driven interrogations, and thereby reducing the usability of the system. The second is that long conversations take longer time to arrive at an accept/reject decision, and likely to create long queues at points of service, such as Airports and guarded doors. In addition, having a line of people behind one person in close proximity may leak private information of the interviewee. Also, others may quickly learn the set of questions and answers that would get them mistakenly authorized, thereby gaining unauthorized access.

We use IRT, which provides the basis for selecting tests from large number of potential questions. Psychmotricans in social sciences and standardized test preparation organizations such as the Educational Testing Services that administer standardized test examinations like SAT, MCAT, GMAT etc. have developed methodologies to measure an examinee's trust or credibility from answers provided to a series of questions. In traditional tests, the ability of the examinee is calculated by adding up the scores of correct answers. Currently, Computerized Adaptive Testing (CAT) that relies on IRT has been used to better estimate an examinee's ability. It has also been shown that the use of CAT/IRT reduces the number of questions necessary to reach a credible estimation of the examinee's ability by 50%. CAT/IRT can be used to control the number and order of questions to be generated based on examinee's previous answers [4, 5].

Our goal in this work is to demonstrate and build an access control system using dialogues of questions and answers generated from a suitable collection of ontologies. Table I shows a sample dialogue that is generated from our research. Our prototype automated IVR system can help immigration enforcement at a border control point making a decision to permit or deny a person asking for entry. Through a dialogue of questions and answers, the interviewee will be assigned a numerical score that will then serve as a threshold in the decision making process. This score is calculated using IRT, which takes into account the correctness of the user's responses and the weight of the individual questions.
The rest of the paper is written as follows. Section II describes an ontological use case, Section III describes the response theory. Section IV describes the system architecture. Section V describes our implementation. Section VI is about experimental results and section VII concludes the paper.

II. Motivating Use Case

In this section, we describe an example ontology used in our work to generate efficient dialogues of questions

TABLE I. A SAMPLE DIALOGUE

	Question/answer	Source	Answer Correct 1 Wrong 0	IRT difficulty	θ (all correct)	θ (all wrong)
System	What is your name	policy	1	1.0	1.0	1.0
User	Dias Kadyrbayev		*Apply policy*			
System	How old are you?	policy	0	1.0	1.0	1.0
User	19		*Apply policy*			
System	I see you have visited Russia, which part?	Ontology - Geospatial	0	1.0	1.3	0.8
User	Dagestan		*Dagestan is a republic within Russia with troubles*			
System	Was there any terrorist attacks while you were there?	Ontology – World Event	1	1.5	1.9	0.5
User	No		*Check world event ontology (ex BBC news Ontology)*			
System	Do you know Dzhokhar Tsarnaev ?	Ontology - FOAF	1	2.0	2.3	0.1
User	Yes	Foaf:knows	*Friend of Dzhokhar Tsarnaev - Brother of Tamerlan Tsarnaev; the Boston bomber.*			
System	Where is he?	Ontology - Geospatial	1	3.0	2.6	-0.5
User	He is in Georgia		*But which Georgia? Is it the country or the US state?*			
System	Are you still attending UMass Dartmouth ?	Dept. of state Visa database	1	3.0	2.6*	-0.5*
User	Yes		*Check if Visa is still valid*			

and answers that are used in assigning a numerical value to an interviewee’s ability or trust level.

Fig. 1 illustrates a class diagram of our under-development ontology for homeland security. The purpose of this ontology is to collect, organize and infer information that can help deterring possible attacks, enforcing strict entry and enabling faster reach to suspects. The ontology defines classes, individuals, properties and relationships using OWL 2 Web Ontology Language (OWL) [9]. The major entities in the ontology are:

- **Person:** defines humans in general and has subclasses like; International Student and Friend.
- **Event:** defines an event that has a location, date, time and type like terrorist attack
- **International Student:** is a person who is on an F-1 or J-1 Visa type
- **University:** defines a university. Some of its current members are MIT and GMU
- **City:** defines a city like Boston
- **Country:** defines a country like USA, Russia, Dagestan, Kazakhstan, etc.
- **State:** defines a state like Massachusetts
- **Visa:** defines visa types like F-1 and J-1 student visas and maybe others.

This ontology represents many kinds of data classes and relationships between these major classes and individuals. For example, we define the “Boston Marathon Bombing” as a “Terrorist Attack” that happened in “Boston”, which is a city in “Massachusetts” state. Another fact is that “Dzhokhar Tsarnaev” is an “Event Character” in the “Boston Marathon Bombing” “Terrorist Attack”. Also we have an “International Student” who is a friend to “Event Character” in the “Boston Marathon Bombing”.

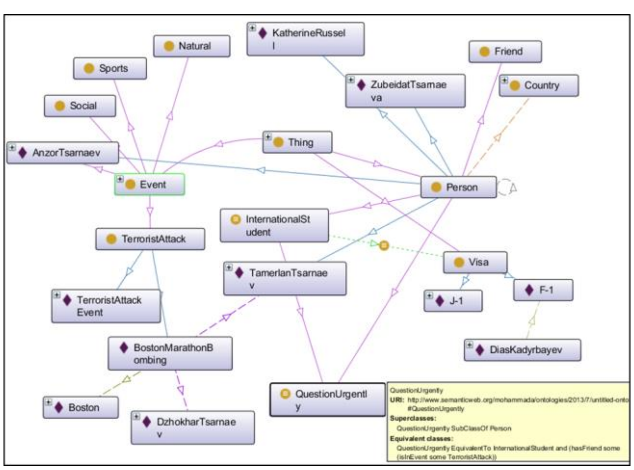

Fig. 1. The Homeland Security Ontology in Protégé

We use this ontology in our work because it serves as a good example showing the strength of our system. First, it shows the possibility of generating valuable questions from asserted or inferred facts. Second, it enables the implementation of the theory under consideration (to be discussed later in the background section) to generate efficient and secure dialogs that are used in: (1) making entry control decisions, (2) assigning numerical values to ability or trust in the shortest time possible and (3) load distribution among interviewers and diverting people for further investigation.

The use of ontology in such an application provides many benefits. The most important amongst them is reasoning. Using a reasoner we are able to derive facts from asserted ones. These facts are used to generate questions to measure the knowledge or ability level of an interviewee on a subject under questioning. In IRT, better item selection and ability estimation happens when a large set of items is available to draw questions from. Using ontology, the large number of derivable facts provides us with the ability to increase the number of questions, and also control the quality and difficulty of questions.

Although there are many reasoners such as FaCT++, JFact, Pellet, RacerPro, we use HermiT [12] in our work. Given an OWL file, HermiT can determine whether or not the ontology or an axiom is consistent, identify subsumption relationships between classes and deduce other facts. Most reasoners are also able to provide explanations of how an inference was reached using the predefined axioms or asserted facts.

One such fact derived from asserted ones in our ontology, is finding the friends that hold a student visa of a person involved in a terrorist attack. To explain this, we have “dzhokhar is friend of Dias”, “Dias is friend of Azamat”, “Dias has F-1 visa”, “Azamat has a J-1 visa”, “dzhokhar is an “Event Character” in the “Boston Marathon Bombing”, “Boston Marathon Bombing” is a “Terrorist Attack”. Thus we infer (using the HermiT reasoner) that Azamat and Dias are the friends of the Boston Bomber and therefore need to be questioned at any entry point. We use this chain of derivations to generate specific questions from them.

Reasoners and the explanations that they provide are very important components in our work to generate relevant and critical questions from ontology that measure knowledge and estimate ability from a response in order to grant access or assign trust. In the example above, the reasoner provided an explanation of the inference using 11 axioms. We use such a number in defining the difficulty of questions generated from such inferences, as

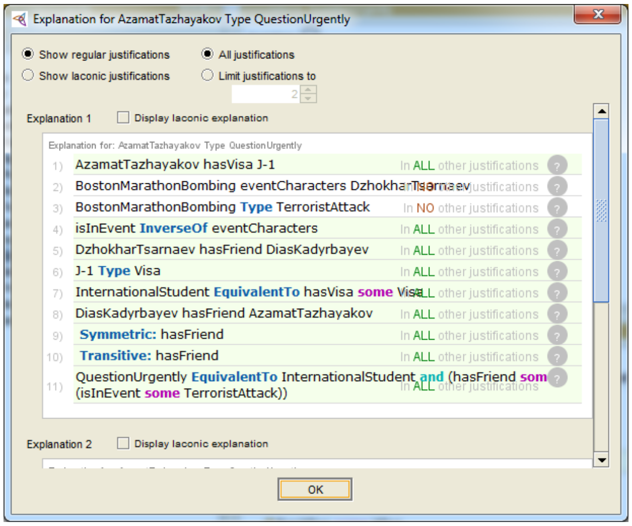

Fig. 2. A sample explanation of an inferred axiom in Protégé using the HermiT reasoner

will be explained in section V. Fig. 2 shows the HermiT reasoner explanation of our inferred fact.

III. BACKGROUND

A. IVR Systems

The main purpose of an IVR system is to interact with humans using a voice stream. An IVR environment consists of a markup language to specify voice dialogues, a voice recognition engine, a voice browser and auxiliary services that allow a computer to interact with humans using voice and Dual Tone Multi-Frequency (DTMF) tones with a keypad enabling hands-free interactions between a user and a host machine [13]. Recently, many applications such as auto attendant, satellite navigation, and personal assistants such as Apple's Siri, Google's Voice, Microsoft's Voice, etc., have started using IVR systems. The IVR language we use is VoiceXML, sometimes abbreviated as VXML [14]. Briefly, Voice XML is a Voice Markup Language (comparable to HTML in the visual markup languages) developed and standardized by the W3C's Voice Browser Working Group to create audio dialogues that feature synthesized speech, digitized audio, recognition of spoken
and (DTMF) key inputs, recording of spoken input, telephony, and mixed initiative conversations.

B. Item Response Theory

IRT, sometimes called *latent trait theory* is popular among psychometricians for testing individuals, and a score assigned to an individual in IRT is said to measure his *latent trait* or ability. Mathematically, IRT provides a characterization of what happens when an individual meets an item, such as an exam or an interview. In IRT, each person is characterized by a proficiency parameter that represents his ability, mostly denoted by (θ) in literature. Each item is characterized by a collection of parameters mainly, its difficulty (b), discrimination (a) and guessing factor (c). When an examinee answers a question, IRT uses the examinee's proficiency level and the item's parameters to predict the probability of the person answering the item correctly. The probability of answering a question correctly according to IRT in a three-parameter model is shown in (1), where e is the constant 2.718, b is the difficulty parameter, a is the discrimination parameter, c is the guessing value and θ is the ability level [3].

$$P = c + (1 - c)\,\frac{1}{1+e^{-a(\theta-b)}} \tag{1}$$

In IRT, test items are selected to yield the highest information content about the examinee by presenting items with difficulty parameter values that are closer to his ability value. This reduces time by asking fewer and relevant questions rather wider range ones while satisfying content considerations such as items or rules that are critical for a decision of access or scoring.

1) IRT parameter estimation

In order to determine the difficulty and discrimination parameters of a test item, IRT uses Bayesian estimates, maximum likelihood estimates or similar methods (MLE) [3, 4]. In the original IRT, an experiment is conducted to estimate these values for each item and at an assumed level of ability for various groups with associated values of IRT parameters using his judgment and experience. Nevertheless, by using our system we can also revise any initial values for these parameters. We model rule attributes as test items and rely on the policy administrator to provide the estimated probabilities.

2) IRT ability estimation

In IRT, responses to questions are dichotomously scored. That is, a correct answer gets a score of "1" and an incorrect answer gets a score of "0". The list of such results consist an item response vector. To estimate the examinee's ability, IRT utilizes maximum likelihood estimates (MLE) using an iterative process involving a priori value of the ability, the item parameters and the response vector as shown in (2). Here, $\hat{\theta}_s$ is the estimated ability within iteration s. a_i is the discrimination parameter of item i, where i=1,2,...,N. u_i is the response of the examine (1/0 for correct/incorrect). $P_i(\hat{\theta}_s)$ is the probability of correct response from (1). $Q_i(\hat{\theta}_s)$ is the probability of incorrect response = 1- $P_i(\hat{\theta}_s)$ [3,4].

$$\hat{\theta}_{s+1} = \hat{\theta}_s + \frac{\sum_{i=1}^{N} -a_i\,[u_i - P_i(\hat{\theta}_s)]}{\sum_{i=1}^{N} a_i^2\; P_i(\hat{\theta}_s)\,Q_i(\hat{\theta}_s)} \qquad (2)$$

Then, the ability estimate is adjusted to improve the computed probabilities with the examinee's responses to items. This process is repeated until the MLE adjustment becomes small enough so that the change becomes negligible. IRT accommodates multiple stopping criteria such as: fixed number of questions, ability threshold or a standard error confidence level. The result is then considered an estimate of the examinee's ability and the estimation procedure stops. The ability or trait usually ranges from -∞ to +∞, but for computational reasons acceptable values are limited to the range [-3, +3].

C. Access Control and XACML

Access control policies specify which subjects may access which resources under some specified conditions [6]. An attribute-based access control policy specifies subjects, objects and resources using some attributes. XACML is an OASIS standard XML-based language for specifying access control policies [7]. In a typical XACML usage scenario, a subject that seeks access to a resource submits a query through an entity called a Policy Enforcement Point (PEP), which is responsible for controlling access to the resource. It forms a request in the XACML request language format and sends it to the a policy decision point (PDP), which in turn, evaluates the request and sends back one of the following responses: accept, reject, error, or unable to evaluate.

IV. USING IRT TO MANAGE AND CONTROL DIALOGUES FROM ONTOLOGIES

Fig. 3 shows the overall architecture of our system. We use derived or axiomatic facts of the ontology to create questions asked by our IVR system. Given that a large number of facts can be derived from our ontology, but only a few questions can be asked during an interview, we use IRT to select the facts that are used to generate questions.

Our questions are automatically created without human involvement by combing English words or phrases such as "Does" or "Is-a" with ones chosen from the ontology of (subject, property, object) triples. The expectation is a dichotomous answer of either (yes, no) or (true, false). The ontological property names such as "is-a", "has-something" are prime candidates for creating true/false questions. Our system transforms the question

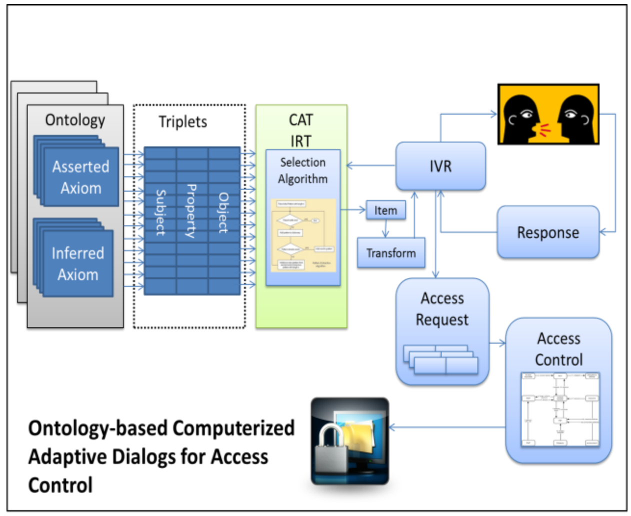

Fig. 3. Ontology-based IVR using IRT

into VoiceXML and plays to the user. Then the system waits for the user's utterance, and if the user provides one, the system's voice recognition software attempts to recognize the input and checks the correctness of the answer. Based on the answer, the IRT estimation procedure either increases a priori ability score or decreases it. The process continues until a predetermined level of ability or accuracy specified according to the application is reached.

Because ontologies produce a large number of facts, it would be impractical to run a dialogue that lasts hours in order to estimate user's ability. In our homeland security ontology uses 167 axioms. The reasoner was able to infer 94 facts raising the total number of axioms and candidate to generate questions to 273.

We use IRT to manage and control dialogue questions generated from a large pool of ontologically derived facts in a way that shortens the length of dialogues while keeping the maximum accuracy in estimating the user's trust. The IRT-based estimated (θ) represents the trust or confidence of the system in the person answering the questions in order to make an access decision.

We have used the OWL annotation property to assign IRT parameters to axioms. Annotations were selected in order to keep the semantics of the original ontology and structure intact. We annotate every asserted axiom in the ontology with IRT parameters, which are: difficulty (b), discrimination (a) and guessing (c). Currently, we assume all asserted axioms have the same default degree of difficulty and discrimination values of 1. The code snippet in Fig. 4 illustrates our annotation using Java with OWL API. An improvement to this approach would be to assign different values for difficulty and discrimination by using domain experts.

```
OWLAnnotationProperty irtDifficultyAP =
df.getOWL
AnnotationProperty(IRI.create("#irt_difficulty"
));
OWLAnnotation irtAnnotation =
df.getOWLAnnotation(
   irtDifficultyAP , df.getOWLLiteral(1.0));
for (OWLAxiom axiom : axioms) {
     OWLAxiom axiom2 = axiom.getAnnotatedAxiom
        (Collections.singleton(irtAnnotation));
     manager.addAxiom(ontology, axiom2);
}
```

Fig. 4. Java code for asserted axiom annotation

We weigh inferred facts more during the estimation process. We are calculating these parameter values from the number of explanation axioms used in each individually inferred fact. Our current scheme of difficulty value assignment is shown in Table II; where higher values or weights are assigned according to the number of explanation axioms used to infer a fact, and consequently the question generated from it is considered to be more difficult than one generated from an asserted fact. Fig. 5 illustrates a code snippet for inferred axiom annotation.

In our current work and for testing purposes we use a default value of “1.0” for discrimination and “0.0” for guessing, which practically neutralizes them leaving the difficulty parameter as the sole factor in estimating ability using equation 2. However, our solution and algorithm are based on the IRT two-parameter model, which relies on the item’s difficulty and discrimination parameters. Fig. 6 shows our algorithm to estimate ability based on equation 2 [3]. Our system estimates the ability of a user after every answer to a question generated from an axiom before selecting and asking the next question. If the ability estimate exceeds the threshold then access is granted. If the threshold is not reached then additional questions are offered. If the estimated ability doesn’t reach the threshold the dialog stops and access is denied. Depending on the application, the dialog might be run again giving a second chance. When the ability estimation again reaches a predefined threshold, the system concludes the dialog and conveys the decision.

TABLE II. IRT DIFFICULTY ASSIGNMENT BASED ON NUMBER OF AXIOMS IN EXPLANATION

Number of explanations	IRT Difficulty	
1	0	Easy
2-3	1	
4-5	1.5	Moderate
8-9	2.5	
>=10	3	Hard

```
Set<OWLAxiom>
inferredAxioms=inferredOntology.getAxioms();
DefaultExplanationGenerator explanationGenerator
=new DefaultExplanationGenerator(
manager, factory, ontology, reasoner, new
SilentExplanationProgressMonitor());
for (OWLAxiom axiom : inferredAxioms) {
    Set<OWLAxiom> explanation =
explanationGenerator.getExplanation(axiom);
//Annotate inferred axioms using the number of
explanation
OWLAxiom tempAxiom =
axiom.getAnnotatedAxiom(Collections.singleton(irt
Annotation));
manager.addAxiom(inferredOntology, tempAxiom);
```

Fig. 5. Java code for inferred axiom annotation

The resultant decision is based on the IRT characteristics of the axiom and not on the number or the percentage of correctly answered questions as in traditional testing. The ability estimate produced by our implementation also comes with a standard error (SE) value that is a measure of the accuracy of the estimate. Equation (3) presents the formula used for standard error calculation [7].

$$SE(\hat{\theta}) = \frac{1}{\sqrt{\sum_{i=1}^{N} a_i^2 P(\hat{\theta}) Q(\hat{\theta})}} \qquad (3)$$

Higher standard error indicates that the estimate is not very accurate, while lower values indicate higher confidence in the estimation. This too can be used as a means to discontinue the dialogue or use an alternate decision method.

V. IMPLEMENTING THE ONTOLOGY-BASED IVR SYSTEM FOR ENTRY CONTROL

Here, we present a prototype of our system showing the major components. It is not yet validated as a deployable system, but it works for the sample use case.

```
Algorithm 1: IRT Ability estimation
Input:a priori theta, Difficulty, Discrimination,
Answer
Output: posteriori theta, standard error
/* calculate theta and standard error*/
for (counter < items.length) do
 itemDifficulty=parseFloat(difficultyArray[i]);
itemDiscrimination=parseFloat(discriminationArr
  ay[i]);
answer=parseFloat(answerArray[i]);
probTheta=calculateProbability(itemDiscriminati
  on,aTheta,itemDifficulty); // equation 1
thetaSplus1= claculateTheta(probTheta, thetaS);
  //equation 2
endfor;
estimatedTheta = thetaSplus1;
return thetaSplus1;
```

Fig. 6. Algorithm for ability estimation in IRT

1) *Voice Platform (Voxeo)*

We use the Voxeo's Prophecy local server as our voice platform for voice recognition and to run the dialogues. Java, Java Server Pages (JSP), and Java Script (JS) are used to implement the architecture modules and to implement IRT procedures used to estimates the user's ability/trust scores.

Voxeo's Prophecy is a comprehensive IVR and standards-based platform [15]. Some of the capabilities integrated into the platform are: automatic speech recognition, speech synthesis (Text-to-Speech), Software Implemented Phone (SIP) browser and libraries to create and deploy IVR or VoIP applications using VXML CCXML. It supports most of server side languages and has a built-in web server.

2) *Item bank*

In our work, we start with ontology, annotate every axiom with an "irt_difficulty" property of value "1". Then we use this ontology in the HermiT reasoner to infer implicit axioms and their explanations. The inferred facts are themselves annotated with "irt_difficulty" property and values calculated by factoring the number of explanation axioms using the schema stated in Table II.

For example, when annotating the inferred fact "*the friends of the Boston Attack Bomber*", which has an explanation that includes 11 axioms shown in Fig. 2, the *irt_difficulty* annotation would be "3.0"; which is the highest value on the scale of IRT difficulty parameter values in Table II. We assume that answering a question generated from a high-valued fact is a difficult task. Consequently, if the answer to a question derived from this fact is correct, the ability estimate would be impacted more positively than a correct, but easy one and more negatively if the opposite happens. An example is the asserted axiom that "Boston is located in Massachusetts". Because this is an asserted fact, it is annotated with value "1.0"; which makes a question generated from it an easy one and thus not affecting the ability estimate greatly.

This process is basically generating the item bank in CAT/IRT terminology. Each item in the item bank contains a question, an answer and IRT parameters. In addition to saving it as ontology in any of the supported formats, this item bank can also be supported by using a more specialized CAT/IRT platform like Cambridge University's Concerto [16].

3) *Generating dialogues from an ontology*

```
<form id="Begin">    <block>
<prompt bargein="true">
   Welcome to the United States. To accelerate
your entry, we will appreciate your responses to
some questions to verify your identity and
eligibility  </prompt>
<assign name="xacmlResource" expr="'point of
entry"/>
<goto next="#Resource"/></block>
</form>
```

Fig. 7. A sample Homeland security VoiceXML greeting form

The conversation starts with a menu in VoiceXML hosted on the local Voxeo Prophecy web server. The voice browser connects to the web server and converts text to speech and speech to text. Fig. 7 shows a sample VoiceXML code.

Fig. 8 shows our algorithm integrating ontology, IVR and IRT. This algorithm was successfully implemented using JavaScript and Java Server Pages (JSP) embedded in VoiceXML pages. The main steps are as follows:

- Load the ontology and parse the XML into Document Object Model (DOM).
- Extract the axiom's triplet (subject, property, object)
- Extract the axiom's IRT difficulty value from the annotation
- Establish a VoiceXML "For" loop that synthesizes a question from string or text values to speech (TTS). The question consists of an auxiliary verb, object, property and subject to test the correctness of an axiom.
- The system waits for a response. If there is one it converts it to text and recognizes it. If it adheres to grammar then a value is assigned as an answer.
- If there was no answer then VXML re-prompts the question up to a programmed number of times. If exceeded then an appropriate VXML is executed.
- The vector of binary answers is used to estimate tIRT ability.
- The loop continues until a threshold of θ or the maximum number of questions is reached.
- The IRT ability estimation algorithm, as illustrated in Fig. 6, takes the variables: answer vector, a priori θ, difficulty, discrimination and calculates a posteriori $\acute{\theta}$.
- If the answer is correct ("yes" or "true"), a value of "1" is assigned. If not, a "0" is assigned.
- The last posteriori $\acute{\theta}$ in the loop is the estimated user's ability θ and can be compared to a threshold value set by an administrator. Access is granted if ($\theta > threshold$) and denied otherwise.

```
Algorithm 2: dialogue access evaluation
Input: a priori theta, Difficulty,
Discrimination, Answer
Output: access control decision
/* make access control decision from
ontology*/
domDocument=parse(ontology); // DOM
subjectArray=getAxiomSubject(axiom);
propertyArray=getAxiomProperty(axiom);
objectArray=getAxiomObject(axiom);
difficultyArray=getAxiomDifficulty(axiom);
/*use voiceXML , JSP to generate dialog*/
for (counter < items.length) do
   <vxml:Prompt> '[auxiliary verb]'
+propertyArray[i] + " " + objectArray[i]
+" "+ subjectArray[i];
   <vxml:Field>= user_utterance;
   response[i] =
Field.voiceRecognition(user_utterance);
   if response[i]= 'Yes' or 'true'
      resultVector[i]=1;
   else
      resultVector[i]=0;
endfor;
theta = IRT_algorithm(resultVector,
difficulty, discrimination,aPrioriTheta);
if theta > thetaThreshold
   permit;
else
   deny;
```

Fig. 8. Ontology-IVR algorithm with IRT

VI. EXPERIMENTAL RESULTS

Our implementation shows that efficient dialogs could be generated from ontologies that have been enhanced with IRT attributes. The successful implementation of the IRT in dialogues of questions and answers shortens the number of questions necessary to reach an accurate estimation of subject's ability, knowledge or trust by at least 50% as it has already been proved by the IRT literature [4, 5]. This reduction of the number of questions necessary to estimate the ability produces shorter dialogs without losing curacy. Also, the use of IRT enables the use of multiple stopping criteria such as: fixed length number of questions or time, ability threshold and standard error confidence interval. The availability of large number of ontology axioms enables generating a set of questions different from another set to be generated immediately after the current user preserving privacy and protecting against question exposure, especially in voice systems. The success of dialog system depends upon multiple timing factors and scalability of supporting multiple users. Our on-going research addresses these two aspects.

VII. CONCLUSION

We have designed and implemented a novel IVR system that can dynamically generate efficient interactive voice dialogs from ontologies for entry control. We have used IRT to generate shorter dialogues between the system and a human speaker. IRT is useful in compensating for inaccurate voice recognition of answers during dialogs or accidental mistakes. Our entry control decisions are made based on an estimation of a level of trust in a subject derived from the importance or relevance of axioms in ontology. The use of IRT also enables the reordering of questions with the purpose of preserving privacy in IVR systems. With the advancement in the fields of mobile, cloud and cloud based voice recognition such systems become important in defence and physical security applications [17, 18, 19].

REFERENCES

[1] M. Ababneh, D. Wijesekera, J. B. Michael, "A Policy-based Dialogue System for Physical Access Control", The 7th STIDS 2012), Fairfax, VA, October 24-25, 2012.

[2] M. Ababneh, D. Wijesekera, "Dynamically Generating Policy Compliant Dialogues for Physical Access Control", CENTERIS 2013 - Conference on Enterprise Information Systems – aligning technology, organizations and people, Lisbon, Portugal. October 23-25, 2013.

[3] F. B. Baker, The basics of item response theory, ERIC Clearinghouse on Assessment and Evaluation, 2001.

[4] D. J. Weiss, G. G. Kingsbury, Application of computerized adaptive testing to educational problems, Journal of Educational Measurement, 21, 361-375, 1984,

[5] H. Wainer, Computerized Adaptive Testing: A Primer, Second Edition, Lawrence Erlbaum Associates Publishers, 2000

[6] M. Bishop, Computer Security: Art and Science, Addison Wesley, 2002.

[7] XACML, OASIS, URL: https://www.oasis-open.org/committees/tc_home.php?wgabbrev= xacml, accessed September 30, 2013.

[8] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993.

[9] W3C, Web Ontology Language (Primer), http://www.w3.org/TR/owl2-primer/ , accessed August 22, 2013.

[10] W3C, SPARQL Protocol and RDF Query Language, URL: http://www.w3.org/2009/sparql/, accessed August 22, 2013.

[11] http://owlapi.sourceforge.net/reasoners.html, accessed August 22, 2013.

[12] http://hermit-reasoner.com/, accessed August 22, 2013.

[13] W3C Voice Browser Working Group, URL: http://www.w3.org/Voice, accessed August 22, 2013.

[14] W3C, Voice Extensible Markup Language (VoiceXML)(VXML), URL: http://www.w3.org/Voice/, accessed August. 22, 2013.

[15] Voxeo web site, URL: http://www.Voxeo.com, accessed August 22, 2013.

[16] Concerto IRT Platform, URL: http://www.psychometrics.cam.ac.uk/page/338/concerto-testing-platform, accessed August 22, 2013.

[17] Microsoft Windows Phone Speech, URL: http://www.windowsphone.com/en-us/how-to/wp7/basics/use-speech-on-my-phone, accessed September 3,2013.

[18] Apple Siri, URL: http://www.apple.com/ios/siri, accessed September 3, 2013.

[19] Google Android Mobile Search, URL: http://www.google.com/mobile/search/, accessed September 3, 2013.