

Chapter 8

AN AUTOMATED DIALOG SYSTEM FOR CONDUCTING SECURITY INTERVIEWS FOR ACCESS CONTROL

Mohammad Ababneh, Malek Athamnah, Duminda Wijesekera,
and Paulo Costa

Abstract Visa, border entry and security clearance interviews are critical homeland security activities that provide access privileges to the geographical United States or to classified information. The person conducting such an interview may not be an expert in the subject area or could be deceived by a manipulative interviewee, resulting in negative security consequences. This paper demonstrates how an interactive voice response system can be used to generate context-sensitive, yet randomized, dialogs that provide confidence in the trustworthiness of an interviewee based on his/her ability to answer questions. The system uses contextual reasoning and ontological inference to derive new facts dynamically. Item response theory is employed to create relevant questions based on social, environmental, relational and historical attributes related to interviewees who seek access to controlled areas or sensitive information.

Keywords: Automated dialog system, security interviews, border control

1. Introduction

Security mechanisms such as guarded gates, border control points and visa issuance counters are implemented to allow access to individuals upon proper authentication and authorization. Legitimacy is usually determined by rules, regulations and/or policies applied by entry control personnel who attempt to ensure that the entry requirements are enforced. Correctly identifying a person may require an examination of an electronic passport, identity card and paper documents in addition to asking the person questions about information contained in the documents.

In order to perform authentication and authorization, an official often conducts an interview in which the aspiring entrant is asked a series of questions based on the specific situation. Ideally, these questions should be relevant, of a reasonable level of difficulty (i.e., neither too difficult nor common knowledge) and should not have been asked previously in similar venues.

The success of interactive voice response (IVR) in auto attendants, satellite navigation devices, personal assistants and mobile applications supported by Apple's Siri, Google Voice and Microsoft's Speech has motivated this research on using interactive voice response systems for access control in visa interviews, entry point interviews, biometric enrollment interviews, resetting passwords and granting access to sensitive resources. Indeed, the goal is to leverage voice technology in automating dialogs used for access control determination.

The use of interactive voice response systems for access control has some limitations. First, most interactive voice response systems have a finite number of pre-programmed conversations. Therefore, the set of questions generated by such a system is the same for every conversation. This may expose the set of questions so that aspiring entrants may arrive with prepared answers. A second limitation is that using a set of random questions from a large pool (typically done to prevent an individual from obtaining the entire knowledge set) may not have the adequate level of difficulty to challenge the aspiring entrant appropriately to determine his/her trustworthiness. A third limitation is that current interactive voice response systems are incapable of discriminating between an individual who knows the subject matter from an individual who correctly guesses the answers. Finally, current interactive voice response technology cannot generate a semantically-coherent sequence of questions that is relevant to the sub-domain that is the focus of the interrogation.

To address the above limitations, this paper proposes an ontological inference-based interactive voice response system that uses item response theory (IRT) to select relevant questions and evaluate trustworthiness. The system uses the access control markup language (XACML) to specify attributes of eligible subjects in the form of an access policy. The proposed system uses an ontology to generate the terms in the access control policy to produce questions. By using an inference engine, a large number of previously-unknown facts about the policy, its attributes and relevant facts are obtained; this provides a large random sample of potential questions [1]. Because the system does not need to store inferred information or save previous question-answer pairs, it can defend against the primary misuse case of human replay attacks. Item response theory is employed to overcome the limitation of selecting relevant questions with acceptable levels of difficulty by creating questions from a large number of attributes present in the policy and ontology.

2. Background

This section describes the theory and technology underlying the interactive voice response system.

2.1 Semantic Web and Ontologies

Mature semantic web technologies, including the Ontology Web Language (OWL), reasoners, repositories and the Simple Protocol and RDF Query Language (SPARQL), are used to model, query and infer information about subjects, objects, relations and attributes [9]. The core facts represented by triples extracted from policy rules and formal sources of information (e.g., law enforcement, Department of State and Department of Homeland Security databases) are stored as an ontology. The representation enables a common and standard format for data usage and sharing. Note that this paper uses a homeland security ontology for demonstration purposes.

The primary advantage of an ontology is the ability to use reasoners [4]. Reasoners are key components of semantic web technologies that can infer implicit facts from existing ones and provide explanations about relationships.

A context-aware application takes into account the context, which includes social, physiological, biometric, environmental, hardware, computational, temporal, activity, identity and location factors [3]. Contextual reasoning involves the inference of contextual information using ontological reasoners. The selection of questions based on item response theory parameters (e.g., difficulty) ensures that the selected questions align within a semantic context. Note that the modeling context enables a reasoner to infer facts relevant to the subject being questioned.

2.2 Item Response Theory

Item response theory, sometimes called latent trait theory, is used by psychometricians to test individuals. An item response theory score assigned to an individual is said to measure the individual's latent trait or ability. Mathematically, item response theory provides a characterization of what happens when an individual meets an item such as an exam or an interview. In item response theory, each individual is characterized by a proficiency parameter θ that represents the individual's ability. Each item is characterized by a collection of parameters: discrimination (a), difficulty (b) and guessing factor (c). When an individual (examinee) answers a question, item response theory uses the examinee's proficiency level and the item parameters to predict the probability that the examinee answered the item correctly. The probability of answering a question correctly in a three-parameter model is given by [2]:

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

where e is the Euler number.

In order to determine the discrimination and difficulty parameters of a test item, item response theory uses Bayesian estimation, maximum likelihood estimation (MLE) and other similar methods [7, 8]. To estimate the examinee's ability, item response theory utilizes an iterative maximum likelihood estimation process involving an *a priori* value of the ability, item parameters and

```

1. <<?xml version="1.0" encoding="UTF-8"?>
2. <Policy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   PolicyId="urn:oasis:names:tc:example:SimplePolicy1"
   RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-overrides">
3. <Description>Homeland Security Example -Secure border entry policy
   </Description>
4. <Target>
5. <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:SimpleRule1"
   Effect="Permit">
6. <Description>A subject who passes a US embassy interview for F-1 visa, will
   be allowed to enter the country temporarily to finish his degree
   Ex:DiasKadyrbayev.</Description>
7. <Target><Subjects>
8. <Subject>
9. <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-
   match">
10. <AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-
   type:rfc822Name"> DiasKadyrbayev </AttributeValue>
11. <SubjectAttributeDesignator
   AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
   DataType="urn:oasis:names:tc:xacml:1.0:datatype:rfc822Name"/
   >
1. </SubjectMatch></Subject></Subjects></Target>
</Rule>

```

Figure 1. Sample XACML policy rule.

response vector:

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

where $\hat{\theta}_s$ is the estimated ability in iteration s ; a_i is the discrimination parameter of item i ($i = 1, 2, \dots, N$); u_i is the response of the examinee (one for correct or zero for incorrect); $P_i(\hat{\theta}_s)$ is the probability of correct response according to Equation (1); and $Q_i(\hat{\theta}_s)$ is the probability of incorrect response.

2.3 Access Control and XACML

Access control policies specify the resources that a subject may access and the conditions under which the access may be granted. This work uses the OASIS XML-based Extensible Access Control Markup Language (XACML) to specify access control policies. In particular, XACML specifies subjects, objects and resources using defined attributes (e.g., verified (ID, password) pair as a subject/object attribute). Figure 1 shows a sample policy rule for the homeland security ontology that allows a person whose `subject-id` attribute value is “DiasKadyrbayev” to enter the United States.

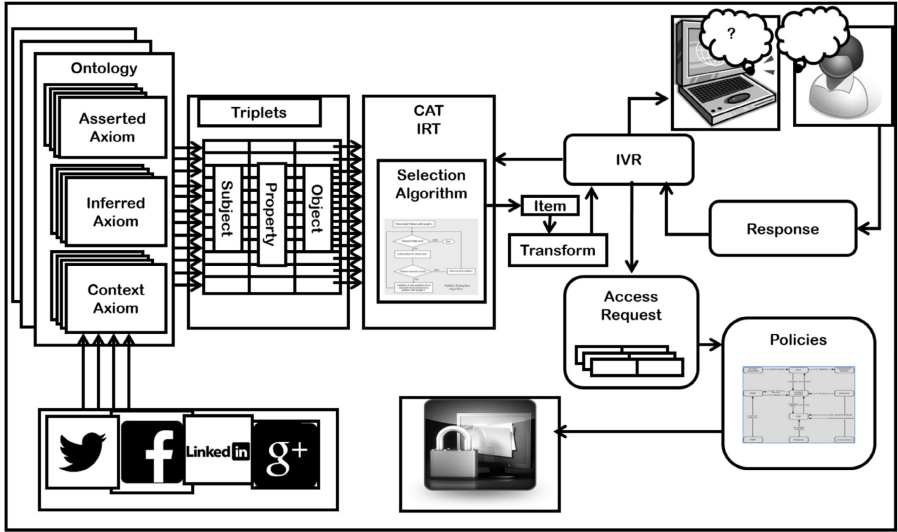


Figure 2. System architecture.

2.4 Interactive Voice Response

Dialog systems provide the means for humans to interact with computer systems. A dialog system uses text, voice and other means to carry out a conversation with a human in order to achieve some objective. Most dialog systems are created with specific objectives in mind and generally involve pre-programmed conversations. An interactive voice response environment incorporates a markup language to specify voice dialogs, 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 via a keypad [9]. VoiceXML is a voice markup language used to create audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF inputs, recording of spoken input, telephony and mixed initiative conversations.

3. System Architecture

Figure 2 shows the overall architecture of the ontology-based interactive voice response system. Axiomatic and derived facts from the ontology are used to create questions asked by the system. Given that a large number of facts can be derived from a context-sensitive ontology, but only a few questions can be asked during an interview, item response theory is used to select the facts that are used to generate questions.

The item response theory module transforms a question into VoiceXML and plays it to the user. The system then waits for the user's response and the system's voice recognition software attempts to recognize the input and check

Table 1. Difficulty assignment based on proof path depth.

Explanations	IRT Difficulty	Level
1	0	Easy
2-3	1	
4-5	1.5	Moderate
6-7	2	
8-9	2.5	
≥ 10	3	Hard

the correctness of the answer. Based on the answer, the item response theory estimation procedure either increases or decreases the *a priori* ability score.

The system uses item response theory to manage and control dialog questions generated from a large pool of ontologically-derived facts in a manner that shortens the length of a dialog while maintaining the maximum accuracy in estimating the user's trustworthiness. When item response theory is not employed, the dialogs tend to be very long or are randomly generated with the possibility of repeated questions.

Another key characteristic of the system is its use of the OWL annotation property to assign item response theory parameters to axioms. Annotations were selected in order to keep the semantics of the original ontology and structure intact. Every asserted axiom in the ontology is annotated with three item response theory parameters, namely discrimination (*a*), difficulty (*b*) and guessing (*c*). Currently, it is assumed that all the asserted axioms have the same default degree of difficulty and a discrimination value of one.

The most important characteristic of the system is that weights are assigned to questions and their answers according to the lengths of the inference or explanation paths. The lengths of the paths are then translated to item response theory difficulty values. Table 1 shows the difficulty value assignment scheme used by the system.

Higher values or weights are assigned according to the number of explanation axioms used to infer a fact. Consequently, such questions are considered to be more difficult than those generated from asserted facts. The item response theory based solution algorithm uses the two-parameter model that relies on the difficulty and discrimination parameters. Figure 3 shows the algorithm used for ability estimation.

After every interactive iteration involving question generation and answering, the item response theory algorithm estimates the ability of the user before selecting and asking the next question. When the ability estimation reaches a predefined threshold, the system concludes the dialog and conveys the decision. Consequently, the decision is based on the item response theory characteristics of the axioms, not on the percentage of correctly-answered questions as in traditional testing.

Algorithm 1: IRT Ability Estimation.

```

Input: difficultyVector, answerVector, aprioritheta
Output: aposterioritheta

/* calculate theta and standard error */
1: for (counter < items.length) do
2:   itemDifficulty = parseFloat(difficultyVector[i]);
3:   itemDiscrimination = "1";           //neutral value
4:   answer=parseFloat(answerVector[i]);
5:   probtheta=calculateProbability(itemDifficulty,
        itemDiscrimination, aprioritheta); // equation 1
6:   thetaSplus1 = thetaSplus1 + MLE(answer, probtheta);
        // equation 2
7: endfor;
8: aposterioritheta = thetaSplus1;
9: return aposterioritheta;

```

Figure 3. Algorithm for ability estimation using item response theory.

4. Involving Context in Dialog Management

This section describes contextual reasoning strategies for selecting questions based on the current context or a previous context. According to the selection criteria, an item is determined only by the item response theory parameters, which have to be in a range of values close to the interim ability estimation $\hat{\theta}$. This can result in questions that are non-homogeneous and unrelated. For example, one question could be: *[is it true (a isBrotherOf b)?]* while the next question could be: *[is it true (Obama isPresidentOf United States)?]*.

Contextual reasoning expands item selection by asking related questions. The concern is not only about the item response theory parameters of a question rendered from an annotated ontology axiom, but also an attempt to continue to ask questions that are related to each other in order to test the user on a branch of knowledge. Compared with the example above, a more relevant series of questions might be *[is it true (a isBrotherOf c)]* followed by *[is it true (b isFriendOf c)]*. Note that the two relations *isBrotherOf* and *isFriendOf* are contextually related by a social relationship context between two people. In order to capture these notions of context, the subject of a previous question is used to generate the next question. Multiple strategies can be used to achieve this objective, such as selecting the next question based solely on subject, solely on property, solely on object or a combination of the three.

A reasoner is used to execute queries in the selection algorithm. A reasoner query using the subject, property or object of a question/axiom is executed to further filter the axioms in the item bank. A contextual reasoning module is incorporated in the architecture to select the axiom elements of the current axiom/question and execute a reasoner query. The addition of this query fur-

ther filters the triple (subject, property, object) that was previously asserted or inferred by the reasoner. The next axiom/question are determined based on a component of the triple.

The axiom parameters of the current question are examined while the voice rendering loop is executing. Subsequently, a query is passed to the reasoner that returns axioms based on the current context. The next question is generated from the newly executed query result and the item response theory parameters that satisfy the item selection criteria for ability estimation and identification.

In the case of historical contextual reasoning, the current context is expanded by saving the user's session questions and answers in an ontology. A reasoner is executed over the closure of the session with the axioms in the item bank ontology. As a result, questions can be asked in subsequent sessions that are related to the questions posed in previous sessions. Selection strategies for multiple sessions include:

- Asking a question related to a question that a user answered incorrectly in a previous session.
- Asking a question requiring deeper knowledge than a correctly-answered question in a previous session.
- Asking a question about personal relationships related to a previous session (e.g., about co-workers, family members or friends).

This capability provides benefits when evaluating an individual multiple times or a group of related people. For example, related attributes are very likely to be encountered during immigration and security clearance interviews. They also provide the ability to detect abnormal changes in user behavior and personality.

4.1 Context Ontology

Historical context is modeled using an ontology that supports reasoning. Figure 4 shows a context ontology developed with the following classes:

- **Item:** Defines a question generated from an axiom triple (subject, property, object) with respect to difficulty annotation and ID annotation.
- **Session:** Defines an interaction session between the user and the interactive voice response system. The session ID is extracted from the interactive voice response.
- **User:** Defines a user. In our application, the user is the “sameAs” an individual in the item bank ontology.

For each question asked, a set of axioms is added to the context ontology. Some examples are: [User_0001 hasQuestion item_0002]; [Item_0001 wasOfferedIn Session_dd552fcdc5fcecf412f96d38818a1c25] and [Session_dd552fcdc5fcecf412f96d38818a1c25 timeDateIs Jun 30, 2009 7:03:47 AM]. The existence of

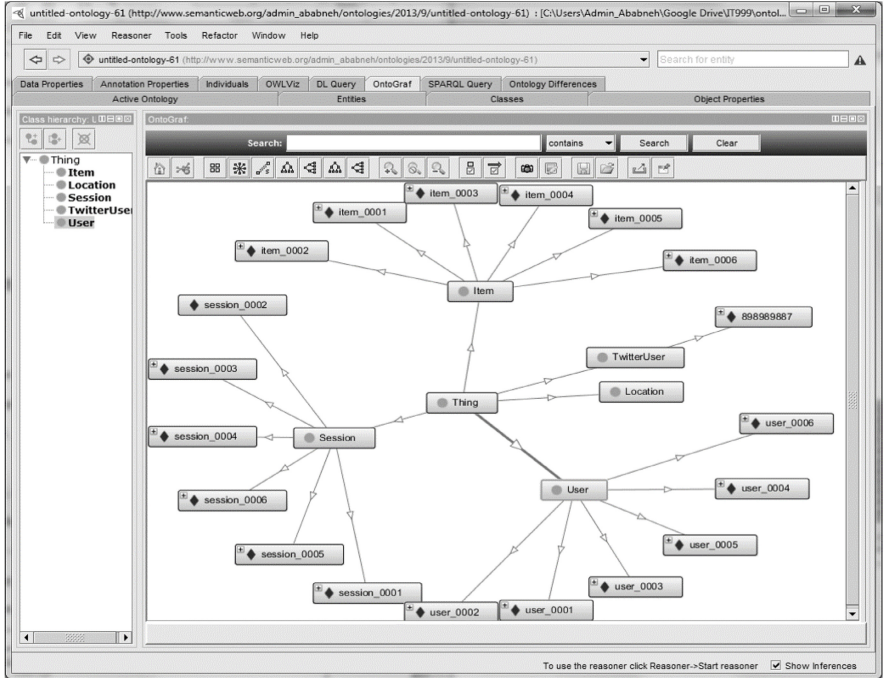


Figure 4. Contextual reasoning ontology.

such axioms makes context-aware reasoning possible. Queries related to questions posed to the current user or to previous users can be executed. This makes it possible to select questions that were previously asked to another user who has a relationship with the current user.

4.2 Contextual Reasoning for Item Selection

The work presented to this point focuses on formal types of policy and knowledge. The axioms must be retained in documents and records. This section discusses how the access control system can be enhanced using knowledge and rules that exist in informal data representations such as social networks (e.g., Facebook, Twitter, LinkedIn and Google+). This is based on the observation that trails, properties, links and photographs contained in social network accounts can be converted to and interfaced with using an ontology to create dialogs used for authentication and authorization. Examples include: Twitter – A isFollowing B; Facebook – B likesPost C; and LinkedIn – C hasContact B and A isGroupMember InformationSecurity.

In the system enhancement, the ontology-based and item-response-theory-supported interactive voice response functionality is augmented with social network information. For example, the system uses Twitter4J APIs [6] associated

```

try{
    long userId=3D97836432, cursor=3D -1;
    IDs ids =3D twitter.getFollowersIDs(userId,cursor);
    // print the user follower IDS=20
    for(long id : ids.getIds())
        System.out.println(id);
    //tweet or status message
    StatusUpdate statusUpdate =3D new StatusUpdate("Twitter from java =
code");
    //tweet or update status
    Status status =3D twitter.updateStatus(statusUpdate);
    System.out.println("status.toString() =3D " + status.toString());
}
catch (TwitterException te){

Followers: 2268362549 921276337 84421394 53510732 1162907539 316653147 1311561259
947407448 967877670 383575552 321475980 809284693 454622361 47586577 528672440
81382232 497434232 41674786 199223028 status.toString() =3D
StatusJSONImpl{createdAt=3DThu Jan 23 15:26:59 EST = 2014, id=3D426451015727329281,
text=3D'Twitter from java code', = source=3D'<a href=3D'http://malek.info'
rel=3D'nofollow'>Malek =
test</a>', isTruncated=3Dfalse, inReplyToStatusId=3D-1, =inReplyToUserId=3D-1,
isFavorited=3Dfalse, isRetweeted=3Dfalse, = favoriteCount=3D0,
inReplyToScreenName=3D'null', geoLocation=3Dnull, =
place=3Dnull, retweetCount=3D0, isPossiblySensitive=3Dfalse, = isoLanguageCode=3Den,
contributorsIDs=3D[J@433ec72e, = retweetedStatus=3Dnull, userMentionEntities=3D[],
urlEntities=3D[], =
hashtagEntities=3D[], mediaEntities=3D[], currentUserRetweetId=3D-1, =
user=3DUserJSONImpl{id=3D97836432, name=3D'Malek S Athamneh',
=screenName=3D'Malekath', location=3D'Fairfax VA', description=3D",

```

Figure 5. Twitter API code and results.

with Twitter. Information extracted from the informal and social knowledge web service is used to populate the context ontology. Figure 5 shows a code snippet and its results, which include the Twitter followers of a specific user and the status of the user. The method `twitter.getFollowersIDs` returns the IDs and the method `twitter.updateStatus` returns the status and location. Axioms such as `[User_0001 isFollowing 1162907539]`, `[81382232 isLocatedIn "Fairfax, VA"]` and `[User_0004 isSameIndividualAs 383575552]` are added to the context ontology.

Leveraging the informal data clearly enhances the questions asked by the system. Indeed, questions can be asked about facts collected from formal interviews, official forms, previous sessions as well as social network attributes.

5. Implementation

This section describes the implementation and the performance characteristics of contextual queries.

5.1 Integrating IVR, Ontology, IRT and Context

The interview begins with a VoiceXML menu hosted on a Voxeo Prophecy web server. The voice browser connects to the web server and converts text-

Algorithm 2: Dialog Access Decision Evaluation.

```

Input:  ontology, contextOntology, user_response,
        difficultyVector, answerVector, aprioritheta
Output: access control decision

/* make access control decision from dialog generated */
/* from ontologies */
1: domDocument = parse(ontology);
2: do
3:   subjectArray = getAxiomSubject(domDocument.axiom[i]);
4:   propertyArray = getAxiomProperty(domDocument.axiom[i]);
5:   objectArray = getAxiomObject(domDocument.axiom[i]);
6:   difficultyVector = getAxiomDifficulty(domDocument.axiom[i]);
7:   while (no_more_axioms)
/* use voiceXML and JSP to generate the dialog */
8:   for (counter < items.length) do
9:     <vxml:Prompt> = "auxiliary verb" + propertyArray[i] + " " +
                    objectArray[i] + " " + subjectArray[i];
10:    <vxml:Field> = user_response;
11:    response[i] = Field.voiceRecognition(user_response);
12:    if response[i] = "yes" or "true"
13:      answerVector[i] = 1;
14:    else
15:      answerVector[i] = 0;
/* contextual reasoning */
16:   Reasoner.Query([subject][object][property], contextOntology);
17:   endfor;
18:   theta = IRT_Algorithm(difficultyVector, answerVector,
                          aprioritheta);
19:   if theta > thetaThreshold
20:     permit;
21:   else
22:     deny;

```

Figure 6. Ontology-IVR algorithm with IRT and context.

to-speech and speech-to-text. Figure 6 shows the algorithm, which integrates interactive voice response, ontology and context. The main steps of the algorithm are:

- **Line 1:** Load and parse the ontology into a Document Object Model (DOM) document.
- **Lines 2–7:** Extract the axiom triple (subject, property, object) and axiom difficulty from axioms and annotations and build separate arrays.
- **Lines 8–17:** Establish a for loop to generate a number of questions synthesized from the axiom triple arrays (subject, property, object and

difficulty) generated in Lines 2–7. An auxiliary verb such as “is” precedes the triples to indicate a true/false question. VoiceXML and JSP are used to render the format.

- **Lines 10–15:** Obtain the user’s response, which is yes/no or true/false. Convert the response to text (0 or 1) and build the response array (vector) used to calculate the ability (θ).
- **Line 16:** Generate questions by contextual reasoning. In addition to the asserted and inferred axioms from the original ontology, a separate context ontology may be used as the target of a reasoner’s query in order to ask the next related question. The reasoner’s query method uses one of the (subject, property, object) triples as a parameter combined with the context ontology to extract the next related axiom/question.
- **Line 18:** Invoke Algorithm 1 to calculate an estimate of the *a posteriori* θ from the *a priori* θ , which represents the ability or trust.
- **Lines 19–22:** The last *a posteriori* θ is an estimate of the user’s ability and is compared with a threshold value set by an administrator. Access is granted if θ is greater than the threshold; otherwise, access is denied.

5.2 Context Performance

Using a context for question generation requires frequent querying of the ontology. To evaluate the performance of question generation, a notional experiment was conducted using synthetic ontology samples that were generated via Java and OWLAPI. SPARQL queries were then executed to return relevant axioms related to specific subjects.

Ontologies were generated for various numbers of students and axioms from a foreign student database. Axioms such as [Student_(randomNumber) isA Student], [Student_(randomNumber) hasVisa F-1] and [Student_(randomNumber) isFromCountry Country_(randomNumber)] were added to produce attributes representative of the Lehigh University Bench Mark (LUBM) ontology [5]. Specifically, an ontology has the following structure:

- There are a number of student individuals of type Student.
- There are 200 countries.
- There are two types of student visas, F-1 and J-1.
- Every student in the ontology has a visa, either F-1 or J-1.
- Every student is a national of one of the 200 countries.
- Every student has a random number of friends (one to twenty) who are also students.

Table 2. Synthetic homeland security ontology statistics.

	HS_100	HS_1k	HS_5k	HS_10K	HS_50K
No. Students	0.1K	1K	5K	10K	50K
No. Axioms (approx.)	2.5K	23K	0.1M	0.2M	1.1M

	HS_75K	HS_100K	HS_200K	HS_250K
No. Students	75K	100K	200K	250K
No. Axioms (approx.)	1.7M	2.2M	4.6M	5.7M

- Every student has participated in a random number of sessions with the system (one to twenty).

Table 2 presents the sample statistics, including the numbers of axioms and their sizes.

To test the performance of contextual reasoning queries, three contextual SPARQL queries were created and executed on the data set. Additionally, two baseline queries were developed for comparison.

- Query15 returns all the predicates and objects participating in an axiom with a designated subject.
- Query16 returns all the predicates and subjects participating in an axiom with a designated object.
- Query17 returns the union of the results of Query15 and Query16.
- Query18 returns all the axioms to provide a baseline for comparison with Query15, Query16 and Query17.
- Query18_100 is a version of Query18 that limits the retrieval of results to 100.

Table 3. SPARQL query execution results.

Query	HS_100	HS_1k	HS_5k	HS_10K	HS_50K	HS_75K	HS_100K
Query15	21	21	44	27	30	29	27
Query16	0	0	9	7	16	2	3
Query17	21	21	53	34	46	31	30
Query18	835	3,018	24,535	239,566	1,199,949	NA	NA

Table 3 shows the SPARQL query execution results after each successful execution. Note that Query18_100 is not included in the table because the

Table 4. SPARQL query execution time (milliseconds).

Query	HS_100	HS_1k	HS_5k	HS_10K	HS_50K	HS_75K	HS_100K
Query15	8	7	13	9	9	10	10
Query16	6	3	8	7	6	7	8
Query17	9	7	10	9	10	9	8
Query18	66	142	1,353	12,133	58,918	NA	NA
Query18_100	20	19	17	18	21	18	23

intent is to show the correctness of the queries. As expected, Query17 shows the union of Query15 and Query16, which is a subset of the total number of axioms returned by Query18.

Table 4 shows the SPARQL query execution time. For a data set containing 10,000 students, Query18 takes approximately twelve seconds to retrieve the results. However, Query15, Query16 and Query17 all require less than 10 milliseconds.

Overall, the three contextual queries perform adequately, returning the expected results in an acceptable period of time. Indeed, the findings suggest that the contextual reasoning queries are executed efficiently and are not expected to cause recognizable delays in question generation that could affect the overall quality of the dialog during an interview.

6. Conclusions

The implementation of an interview system using ontologies, item response theory and contextual reasoning is certainly feasible. The use of ontologies and reasoning is critical to developing dialogs with items that are differentiated quantitatively. Item response theory provides a means to quantitatively characterize dialog items and to measure user trustworthiness and ability. Contextual reasoning provides the means to select the most appropriate questions quantitatively as well as questions that are semantically-relevant to the domain and subject of focus. The enhancement of contextual reasoning with social media information effectively supplements policy and ontology formal attributes with static and dynamic social attributes.

The paper also demonstrates that social media information is very useful for driving dialogs in interviews. However, the information was used without any analysis. Sentiment analysis is a growing area of research that attempts to predict trends in the inclinations or feelings of groups of people towards life issues. By leveraging social media, it is possible to deduce an individual's sentiments about a variety of issues, especially those related to national security. Our future research will focus on implementing a sentiment analysis module that builds on the social attributes module. The use of social media raises some legal concerns. At this time, we assume that individuals provide consent to use public social media information; however, a comprehensive evaluation of the

legal ramifications is required before social media can be used in real interviews of individuals who seek access to controlled areas or sensitive information.

References

- [1] M. Ababneh and D. Wijesekera, Dynamically generating policy compliant dialogs for physical access control, *Proceedings of the Conference on Enterprise Information Systems*, pp. 361–370, 2013.
- [2] F. Baker, *The Basics of Item Response Theory*, ERIC Clearinghouse on Assessment and Evaluation, University of Maryland, College Park, Maryland, 2001.
- [3] B. Beamon and M. Kumar, HyCoRE: Towards a generalized hierarchical hybrid context reasoning engine, *Proceedings of the Eighth IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 30–36, 2010.
- [4] Information Systems Group, HerMiT OWL Reasoner, Department of Computer Science, University of Oxford, Oxford, United Kingdom (hermit-reasoner.com).
- [5] Semantic Web and Agent Technologies Laboratory, SWAT Projects – The Lehigh University Benchmark (LUBM), Department of Computer Science, Lehigh University, Bethlehem, Pennsylvania (swat.cse.lehigh.edu/projects/lubm).
- [6] Twitter4J, Twitter4J (twitter4j.org/en/index.html).
- [7] H. Wainer, N. Dorans, R. Flaugher, B. Green and R. Mislevy, *Computerized Adaptive Testing: A Primer*, Routledge, New York, 2014.
- [8] D. Weiss and G. Kingsbury, Application of computerized adaptive testing to educational problems, *Journal of Educational Measurement*, vol. 21(4), pp. 361–375, 1984.
- [9] World Wide Web Consortium, OWL 2 Web Ontology Language Primer, Massachusetts Institute of Technology, Cambridge, Massachusetts (www.w3.org/TR/owl2-primer), 2012.