Speech acts and Tokens for Access Control and Provenance Tracking

Fabian Neuhaus
National Center for Ontological Research

Bill Andersen
Highfleet, Inc.

***Abstract*—In many applications, ontology-based technologies will be only only be successful if they support access control and provenance tracking. In this paper we present a novel approach to implementation of both access control and provenance in deductive information systems. A key feature of our approach is the explicit representation of speech acts as well as sentence tokens that are used to encode propositions. These are used to define *SupportedBy*, a kind of entailment relationship between sentence tokens and propositions. User queries are phrased in terms of the SupportedBy relationship and augmented by user-dependent security and provenance constraints. We note that the introduction and treatment of SupportedBy makes the resulting logic an instance of a *labeled deductive system*, as developed by Gabbay.**

I. Introduction

To date, ontology-based technologies have been so far applied most successfully in domains like biological research where the available knowledge meets two important requirements. First, there is a network of trust among users and builders of the knowledge base that it represents a true picture of reality. Second, the knowledge in the knowledge base is open in the sense that anybody who is using the system can access all the information in the knowledge base.

However, many potential applications ontology-based systems require multi-level security access control along with 'need-to-know' restrictions. Examples include the management of the information exchanges within an engineering production network, patient data within a hospital management system, and data analysis within an intelligence agency. In addition, any application of knowledge representation technologies such applications would require reasoning with information that might turn out to be wrong, either by mistake or by ill intent. The available information might even turn out to be logically inconsistent. Within this context, it is almost as important to keep track of who provided a piece of information as keeping track of the information itself. This enables to evaluate the quality of a given information item by checking its consistency with information provided by independent sources.

Provenance tracking comes in two flavors: *hearsay tracking* and *IT processing tracking*. Hearsay tracking is concerned with the chain of 'retellings' of a piece of information before it is entered into an information technology (IT) system; e.g., in 'Novak reported that a senior official claimed that Plame suggested that Wilson travels to Niger' the basic proposition 'Wilson travels to Niger' is embedded in three layers of 'retellings'. IT processing tracking is concerned with how information is processed within a given set of IT systems. After the information is entered into an information system, it might be processed in various ways. It can be copied, recoded, or used for automatic reasoning, among others. It is important to track these processes because they can potentially cause false verifications.

In this paper we discuss the features of an ontology language that can support provenance tracking. This approach, which is based on work described in [8] and [1], has already been implemented successfully in Highfleet's XKS deductive database system. This paper clarifies the relationship of linguistic tokens and speech acts in the analysis of provenance tracking, and widens the scope of the analysis to cover hearsay tracking. As in previous work, we take a logical-ontological approach that considers (1) the entities required for an adequate accounting of access control, (2) provenance in information systems, and (3) the logical machinery that is needed to get the intended result. Hence, we will not discuss in this paper the details of the implementation.

In the next section we provide an extended example from the intelligence community that illustrates the kind of problems we intend to address in this paper. Afterward we discuss the ontological categories involved in our solution. In the last section we present a first draft of our account.

II. Example

In this paper we discuss our approach with the help of the following scenario. Assume that an Afghan source of a U.S. intelligence agency reports that Al Qaeda has obtained a nuclear weapon. The information is represented in the knowledge repository A and classified as top secret. The information about the supposed location is shared with another US agency, but the source of the information is not shared. The second agency stores the information within their knowledge repository B and classifies it as secret. Assume further, that in the same timeframe the New York Times (NYT) reports that Maulana Masood Azhar claimed in an interview that Al Qaeda has obtained a nuclear weapon. This is recorded in the knowledge repository C, which contains information collected from newspapers and other publicly-available sources. As a result, all three repositories contain (in some sense) the same information, namely that Al Qaeda has obtained a nuclear weapon. However, it is classified differently in the knowledge repositories A, B, and C (as top secret, secret, and unclassified,

respectively). To further muddy the water, assume that repository B also contains reports from other sources that claim that if Al Qaeda has a nuclear weapon then it has obtained it from Pakistan; and that in addition no Pakistani nuclear weapon is missing. Thus, repository B contains conflicting information.[1]

Assume an analyst queries an information system with access to all three knowledge repositories with the following request: *Provide all independent records that support that Al Qaeda possesses weapons of mass destruction.* For the sake of simplicity, let's further assume that the knowledge repositories contain no other relevant information. This scenario provides several challenges: (i) The correct response from the system depends on the clearance of the analyst. If the analyst has no access to classified information, he should receive one answer, namely the one from the NYT report in repository C. The fact that the information provided by the Afghan source is classified should not prevent the analyst from accessing the information based on news reports, although in some sense it is the *same* information. (ii) If the analyst has access to top secret information, the system should provide two answers and not three: the NYT report and the original record in repository A. The system should not provide the record in knowledge base B, since it is based on the one in repository A, and thus does not provide independent verification. (iii) Inferring that the report by the Afghan source and the NYT article is about weapons of mass destruction requires logical reasoning with content embedded in some additional information about provenance. (iv) The available information is logically inconsistent; a fact that seems to render the classical entailment relationships useless.

We have addressed the first two challenges in [1]. In this paper we extend our solution to hearsay provenance tracking and address the problem of inconsistent information in more detail.

III. Ontology of access control and provenance

Our approach to a theory of access control is ontological rather than procedural.[2] By first examining and fixing the relevant kinds of entities involved in access control and provenance, we hope to provide a firm foundation for an evolving formal theory for handling these phenomena in information systems.

A. Information, Proposition, Sentence Types, and Tokens

According to the U.S. government the object of access control is information [9]. While an analysis of the ontological nature of information is beyond the scope of this paper, we are convinced that according to any reasonable account of information (e.g., [12]) a unit of information is an 'abstract entity' in the same sense that integers or geometric shapes are abstract entities. That is, they do not have a spatio-temporal location and are not participating in causal interactions that shape our physical world. Based on this view, it is hard to imagine how we might control directly access to anything abstract. What we can control, though, is access to physical entities that *encode* information. For example, it is impossible to lock a piece of information in a safe for the same reasons it is impossible to lock up the integer 3. We are able, however, to lock up a paper document that encodes the information. In the case of IT systems the computational mechanisms have causal influence over objects that are encoded ultimately as patterns of electrons or some other physical mechanism. We argue that the objects of access control are thus spatio-temporal objects that participate in the causal structure of information systems.

Although we are confident that our approach can be extended to information encoded in images, video, audio and other like forms of common digitally encoded media, but in this paper we will focus on information systems dealing with propositions encoded in formal language expressions. A *proposition* is a unit of information that is either true or false. A well-formed expression of a (formal) language that expresses a proposition is a *sentence type*. We note that 'formal language' is not intended to be restricted to logical languages but is intended to cover any kind of syntax including graphical notations, tables, tree structures, and barcodes.

Since sentence types and propositions are abstract entities they cannot be objects of access control and provenance tracking for the reasons given above. In contrast, *sentence tokens* are physical entities that instantiate sentence types [13]. The same sentence type can be instantiated by a large range of physical objects; a sentence token on a printed newspaper is a distribution of ink, in the case of a spoken sentence the token is a complex movement of air, and in the case of information systems the tokens are arrangements of electric charges in a chip.

Different tokens of the same types might not only differ with respect to the kind of material they consist of and other physical qualities, but, more importantly for our purposes, different tokens of the same type can differ with respect to their security properties: one encoding of a proposition P might be unclassified while another encoding of the same proposition one is classfied.

A sentence token might come into existence by accident. If you spill your coffee on a sheet of paper and it reads "It was the best of times", then this distribution of coffee on this sheet of paper is an instance of the sentence type. However, usually sentence tokens are brought into existence by a person in an attempt to communicate with somebody else, a *speech act* [2], [11]. A speech act is a kind of intentional act performed by a person (the speaker) typically involving one or more listeners, a sentence token, a proposition, and the *illocutionary force* of the speech act. For example, utterances of the assertion 'You are late', the question 'Are you late?' and the command 'Be late!' involve the same proposition but vary in their illocutionary force – the first makes a statement about reality, the second seeks verification, and the third seeks to bring about the truth of a proposition. While these examples might suggest that illocutionary force is aligned with syntactic

[1]For the sake of simplicity, we do not treat time explicitly and just assume that the statements are valid during the same time period.

[2]The Bell-La Padula security model is one example where secure states of a system are defined by a state machine model [3].

distinctions in English, this is not the case. E.g., the utterance of 'I'll be back' might be a promise, a prediction, a warning, or a threat – depending on the circumstances and the intentions of the speaker. This example shows that while under normal circumstances the proposition of a speech act is straightforwardly encoded in the sentence token, the illocutionary force might be harder to determine. For the sake of simplicity, we will in the following widely ignore the differences between the illocutionary forces of the various types of speech acts and treat them either as assertive speech acts or as queries. For example, promises, warnings, and threats will all be treated equally as assertions.

We are concerned with speech acts for two reasons. First, if a sentence token in an IT system is the result of an entry by a human, then the sentence token is the result of a speech act. Second, we need to deal with sentence tokens that encode propositions about speech acts. In our example an analyst has read in the NYT that Maulana Masood Azhar claimed that Al Qaeda owns a nuclear weapon, and creates a corresponding entry in a knowledge repository. Creating this entry is a speech act by the analyst. The token in the knowledge repository does encode the proposition that the NYT performed a reporting speech act. The propositional content of the speech act by the NYT is that Maulana Masood Azhar has performed another speech act, namely an announcement. The proposition of that last speech act is that Al Qaeda owns a nuclear weapon. Thus, the propositions of all three speech acts are nested within each other.

B. Manipulation of tokens in IT systems

By IT system we mean a physical object that is capable of (1) accepting information encoded in tokens of some appropriate language and (2) accepting and responding to queries posed as tokens in some appropriate language with the result being a *release* of tokens encoding the query response. In this paper we are interested in IT systems with access control, that is systems that allow access to stored information only through specified processes and through no other means.

On a token-based view of access control, the policies that guide whether a given token can be released by an IT system is based on its access control properties. Thus, we must account for the causal history of tokens in an information system from the moment that information bearing tokens enter a system to when (other) tokens are released from the system. This causal history will take the form of a chain of events (copying, synthesis, and recoding) that make new tokens from old ones. Depending on the type of event, properties relevant to access control will need preservation.[3]

In this paper we consider a security labeling system that consists of a totally-ordered set of *levels* L and a set of partially ordered *compartments* C. Each token is assigned a security level and a (potentially empty) set of compartments. Security levels express the sensitivity of a given piece of information. Compartments are used to limit access channels independent of the security levels. The partial order on the set of compartments ranks the compartments along their specificity (e.g., the compartment *Al Qaeda* would be more specific than the compartment *terrorist group*). Ontologically speaking, security levels and compartments are social artifacts that are dependent upon a community of agents that mutually agrees to the storage and access of tokens using the labeling system.

[3]We discuss IT systems and their boundaries, as well as the the manipulation of tokens within IT systems in greater length in [1].

IV. Formalization of access control and provenance

A. The representation in a formal language

In this section we will sketch an axiomatic approach that allows us to reason under multi-level security access control and enables provenance tracking. We are not suggesting that the logical language below is supposed to be used within a knowledge repository, nor do we suggest that the end users of the system shall be confronted with such a language. Our main concern is that the features of the implemented knowledge representation language enable queries and logical reasoning in a way that supports access control and provenance tracking as described here.

As mentioned in the introduction, Highfleet has already implemented a system with these features successfully. However, the goal of this paper is not describe a specific implementation or to discuss how the approach we are suggesting can be implemented efficiently. Our goal is just to outline the underlying logical-ontological approach. For this reason we just assume that we have a reasoner that supports a very expressive language, at least as expressive as IKRIS Knowledge Language (IKL) [5], [6] extended by two modal operators: $\Diamond$ is read as 'it is logically possibly true that' and $\Box$ is read as 'it is logically necessarily true that'.[4] IKL is an extension of CLIF which itself is the interchange format of the ISO standard Common Logic [7]. CLIF differs from many first-order languages by not assigning a fixed arity to its predicates and by adding sequence variables to the language. In the following we will use x, y, z as ordinary first-order variables and s, s_1, s_2 as sequence variables – variables that range over finite sequences of objects.

One basic idea of our approach is to treat tokens that reside in the repositories and the speech acts they encode as first class citizens in the domain of discourse. In this way the so-called 'metainformation' about security and provenance can be treated in the same logical framework as regular object-level information. Let's return to the example from the introduction. The knowledge repository A of the first agency contains a token that expresses *Source007 asserted on October 20th, 2011 that Al Qaeda owns nuclear weapons*. The most important aspects of assertive speech acts are its speaker as well as the proposition that is asserted. The example

[4]The intended semantics is the following: $\Box A$ is true if and only if A is logical truth of classical first-order logic. $\Diamond A$ is true if and only $\sim A$ is is not a logical truth of classical first-order; i.e. A is satisfiable. The details of the extension of IKL by these modal operators are beyond the scope of this paper.

includes also the date of the speech act. Potentially other relevant information about the speech act might be available (e.g., its location or the listeners that participated in it). This 'metadata' about the speech act needs to be distinguished from the 'metadata' about the token that encodes the speech act itself. Examples for 'metadata' about tokens include the type of the token (e.g., record, audio file, picture), the name of the repository where the token resides, the security classification of the token, its security compartments, and the person who created the entry in the system, the date when the entry was created, a list of people who accessed the information in the system, and so on.

This 'metadata' can be represented in the same framework as the 'normal' data in the following way, where 'token001' is a name of the record that resides in the repository A:

Tok1
Record(token001) &
ResidesIn(token001) = repository_A &
ClassifiedAs (token001 top_secret) &
Compartment (token001 alQaeda_cmpt) &
Compartment (token001 proliferation_cmpt) &
CreatedBy(token001) = agent1234 &
PropositionalContent(token001) =
(that ($\exists x$ *(AssertionAct(*x*)* &
Speaker(x *source007)* &
Date(x*) = 20.10.2011)* &
PropositionalContent(x*) = (that(*
$\exists y$ *(Owns (alQaeda* y*)* & *NuclWeap(*y*)))))*

We use IKL's mechanism for expression of propositions – the that-operator. It is applied to a formula and the result is a name that refers to a proposition. Its logical counterpart in IKL is a syntactic variant of a truth-predicate: if p is a proposition, then the formula (p) is the assertion of the proposition. Thus, $(A \leftrightarrow ((\text{that } A)))$ is a logical truth in IKL, for any formula A.

In our example, the agency that owns repository A shares the information with another agency. As a result a 'write down' token is created within repository B; that is the propositional content of the speech act encoded in token001 is preserved, but the additional information about the speech act is removed. As a result the newly created token is reclassified as secret. The information about the token in repository B can be represented in the following way:

Tok2
Record(token002) &
ResidesIn(token002) = repository_B &
ClassifiedAs(token002 secret) &
BasedOn(token002, token001) &
ResidesIn(token001) = repository_A &
PropositionalContent(token002) =
(that ($\exists x$ *(AssertionAct(x)* &
PropositionalContent(x) = (that(
$\exists y$ *(Owns (alQaeda* y*)* & *NuclWeap(*y*)))))*

The fact that the entry in knowledge base B originated from repository A is expressed explicitly by asserting that token002 is based on token001 and that token001 resides in the repository A. Using the 'BasedOn' relationship in this way enables provenance tracking across the knowledge repositories, and enables a reasoner to detect that token001 and token002 do not provide independent verification of the information concerning Al Qaeda's access to nuclear weapons. This example points to a further advantage of our approach – namely that it provides a principled way of performing 'write-down' operations, enabling more flexible sharing of information without compromising sensitive meta-information. Such operations are typically not allowed by traditional security models, e.g., [3].

The entry based on the NYT report is distinguished from the previous examples by an additional layer of indirectness: the token encodes a proposition about a speech act about a speech act. Each of the speech acts has a propositional content and a speaker; in this example the dates of the speech acts are provided as well.

Tok3
Record(token003) &
ResidesIn(token003) = repository_C &
ClassifiedAs(token003 unclassified) &
PropositionalContent(token003) =
$\exists x$ *(AssertionAct(x)* &
Speaker(x nyt) &
Date(x) = 23.10.2011 &
PropositionalContent(x) =
(that ($\exists y$ *AssertionAct(y)* &
Speaker(y MasoodAzhar) &
Date(y) = 22.10.2011 &
PropositionalContent(y) = (that(
$\exists z$ *(Owns (alQaeda* z*)* & *NuclWeap(*z*)))))*

B. The support relationship

We now add another relationship "SupportedBy" between a proposition and a sequence of zero or more records. The goal of this relation is to capture not only the propositional content that is captured in one record, but what is logically entailed by the sequence of these records. One problem we need to address is that in the framework of a classical logic a contradiction logically entails any proposition. Assume we have an ontology-based information system with a classical reasoner. In our example, the knowledge base B contains records that encode the following propositions: (i) Al Qaeda owns a nuclear weapon; (ii) if Al Qaeda owns a nuclear weapon, then Pakistan is missing it, and (iii) the Pakistanis are not missing a nuclear weapon. If we were to provide these propositions in an unaugmented way to this system, the reasoner would 'use' these contradictory assumptions to prove any query – and thus the IT system would become useless. Our goal is to enable limited reasoning with contradictory information, but to prevent the system from 'exploding'.[5] This

[5]This goal is the driving force behind the development of paraconsistent logics. Since we are defining a (object language) relationship between tokens and propositions our goal is slightly different than the one in paraconsistent logic which is concerned with the (meta language) entailment relationship. The "SupportedBy" could be briefly characterized as a paraconsistent variant of the strict implication with a closure on embedded propositions.

is achieved with the help of the two modal operators $\Diamond$ and $\Box$ introduced above.

Instead of SupportedBy((that A), s) we write A[s] as a shorthand. In particular, we write A[ ] to express that A is supported by the empty sequence. We axiomatize the SupportedBy relationship recursively with the following axiom schemata:

Ax1 $(\mathit{Record}(x)\ \&\ \Diamond(\mathit{PropositionalContent}(x))) \rightarrow \mathit{SupportedBy}(\mathit{PropositionalContent}(x), x)$

Ax2 $A \rightarrow A[\,]$

Ax3 $(A[s_1]\&B[s_2]\&\Diamond(A\&B)) \rightarrow (A\&B)[s_1\ s_2]$

Ax4 $(A[s]\&\Box(A \rightarrow B)) \rightarrow B[s]$

Ax5 $(\Diamond A\ \&\ (\exists x((\mathit{AssertionAct}(x)\& (\mathit{PropositionalContent}(x) = (\mathit{that}\ A))))[s]) \rightarrow A[s]$

Ax1 expresses that every record supports its (own) propositional content – under the condition that assertion of the propositional content is possibly true. Further, every proposition that is already known to be true is supported by the empty sequence (Ax2). According to Ax3 the following holds: if a proposition A is supported by a sequence of records s_1 and a proposition B is supported by a sequence of records s_2 and (A & B) is possibly true, then the proposition (A & B) is supported by the sequence that is the result of concatenating s_1 and s_2. Note that if A and B are logically contradictory, it is not possible that (A & B) is true; thus in this case A[s_1] & B[s_2] do not imply (A & B)[s_1 s_2]. Without this constraint a sequence of assertions of contradictory information would support every proposition because, as discussed above, in classical logic a logically false formula will entail any formula. Ax4 expresses the following: if the sequence s supports a proposition A and A necessarily implies B, then the sequence s also supports the proposition B. The axiom ensures that a sequence of records does not only support a conjunction of their propositional contents but also the logical consequences of the propositions. Ax5 ensures that a token does not only support the proposition it encodes but also all propositions that are embedded in that proposition – provided that they are logically possible.

We made a few simplifications in these axioms. First of all, we axiomatized SupportedBy based on sequences of tokens. Sequences that consist of the same components in different order are different sequences; e.g. (token001 token005) and (token005 token001) are two different sequences. Consequently, an IKL reasoner will consider them as different answers to a query. However, for SupportedBy the order of the sequence elements does not matter, any permutation is as good as another. Further, the approach delivers sequences that contain tokens that are not necessary to support the proposition. For example, the answer (token001 token005) would be a valid answer to query Que1, in spite of the fact that token001 supports the proposition on its own and token005 does not contribute anything to the answer. Thus, the axioms as presented above would deliver redundant answers. It is possible to avoid these problems, but for the sake of brevity we present a simplified account.

C. Reasoning with SupportedBy

The support relationship is used to enable queries for information that support a given hypothesis. In the rest of this section we will show how that works with the help of the example from the introduction. However, within this section we will ignore that Tok2 is based on Tok1; this will be the subject of the next section. Let's assume that the system has access to an ontology that either contains or logically entails the background information Bgnd1: A nuclear weapon is a weapon of mass destruction (WMD).

Bgnd1 $\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))$

In our example, the analyst is interested in the question whether Al Qaeda possesses WMD. For starters, we can represent the query 'Find all sequences of records that are supporting the proposition Al Qaeda owns WMD.' in the following way:

Que1 $\exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{WMD}(x))[?s]$

Note that IKL itself does not provide any convention to express queries; hence, we use question marks in front of variables to mark variables to be bound by a reasoner.

When the analyst enters the query Que1 into the system, it tries to find a sequence of tokens that enables it to prove the query. For example, the system would try to prove that token001 supports this proposition. Example1 shows how a proof could look like.

Example1

1) $\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))[\,]$
2) $\exists x(\mathit{AssertionAct}(x)\ \&\ \mathit{Speaker}(x\ \mathit{source007})\ \&\ \mathit{Date}(x) = \mathit{20.10.2011})\ \&\ \mathit{PropositionalContent}(x) = (\mathit{that}(\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\&\mathit{NuclWeap}(y)))[\mathit{token001}]$
3) $\Box((\exists x(A\ \&\ B\ \&\ C\ \&\ D)) \rightarrow \exists x(A\ \&\ D))$
4) $\exists x(\mathit{AssertionAct}(x)\ \&\ \mathit{PropositionalContent}(x) = (\mathit{that}(\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\&\mathit{NuclWeap}(y)))[\mathit{token001}]$
5) $\Diamond(\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\&\mathit{NuclWeap}(y)))$
6) $\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{NuclWeap}(y))[\mathit{token001}]$
7) $\Diamond(\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))\& \exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{NuclWeap}(y)))$
8) $(\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))\ \& \exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{NuclWeap}(y)))[\mathit{token001}]$
9) $\Box((\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))\ \& \exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{WMD}(y))) \rightarrow \exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x)))$
10) $\exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x))[\mathit{token001}]$

Line 1 of the proof is an immediate consequence of Bgnd1 and Ax2. Line 2 follows from Tok1 and Ax1. Line 3 is a modal theorem schema, and the proposition of line 2 matches the antecedent. Thus, line 3 in combination with Ax4 can be used to remove the information about the speaker and the date from line 2, the result is line 4. Lines 5, 7, and 9 are theorems

under the intended interpretation of the modal operators. Lines 4, 5, and Ax5 give rise to line 6. Lines 1, 6, 7, and Ax3 entail line 8 of the proof. Line 8, 9, and Ax4 entail line 10. Q.E.D.

Example1 shows that (token001) (the sequence consisting only of token001) is one possible answer to query Que1. In a similar fashion one can prove that (token002) and (token003) answer the query. While Example1 is admittedly rather simple, it is sufficient to show how these proofs work and what role the axioms play: the 'background information' that is provided to the system as truths (e.g., nuclear weapons are WMD), lead to SupportedBy-statements with an empty sequence by axiom Ax2. Formulas that express the content of records (like Tok1) lead to SupportedBy-statements via axiom Ax1 that contain lists with only one element. In our example we use these axioms only once each, but in more complex examples one would have to use these axioms repeatedly. The resulting SupportedBy-statements can be combined with more complex ones with the help of axioms Ax3. The role of Ax4 is to ensure that consistent lists of tokens support all logical consequences of their propositions. If a proposition is embedded in another proposition, then Ax5 (in combination with Ax4) allows us to show that the former proposition is supported by the same sequence of tokens as the latter.

Example2

1) $\forall x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x)) \rightarrow \mathit{Misses}(Pakistan\ x))[\mathit{token004}]$
2) $\sim\exists x(\mathit{Misses}(Pakistan\ x)\ \&\ \mathit{NuclWeap}(x))[\mathit{token005}]$
3) $\sim\exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x))[\mathit{token004\ token005}]$

In Example1 the proposition in the last line is only supported by one token, but a proposition can be supported by an arbitrary long list of tokens. Let token004 encode the proposition 'if Al Qaeda owns a nuclear weapon, then Pakistan misses it', and let token005 encode 'Pakistan is not missing a nuclear weapon'. By applying Ax1 we get the first two lines of Example2. They in combination with Ax3 and Ax4 entail the third line: an example for a proposition supported by two records.

Note that it is not possible to combine the last lines of Example1 and Example2 with Ax3, because $\Diamond(A\ \&\ \sim A)$ is not provable, for any given formula A and any given set of consistent assumptions. This is an example how the axioms of the SupportedBy relationship block unwanted reasoning with inconsistent information.

So far, in all examples that we discussed the provenance of the information has been ignored. So what is the benefit to represent the speech acts explicitly within the formulas? First of all, the analyst does need to know who provided the information and whether it is hearsay or the result of direct observation. In addition, it allows us to support queries that mix 'normal' queries with 'metainformation' about security classification and provenance. For example, the analyst might ask the following additional query: *Find all top secret records that involve an assertion by Masood Azhar that entail the existence of nuclear weapons.* This query can be represented as follows:

Que2 $\mathit{Record}(?x)\ \&$
$\mathit{ClassifiedAs}(?x\ \mathit{top\_secret})\ \&$
$(\exists y(\mathit{AssertionAct}(y)\ \&\mathit{Speaker}(y\ \mathit{MasoodAzhar})\ \&$
$\Box((\mathit{PropositionalContent}(y)) \rightarrow$
$\exists z\mathit{NuclWeap}(z)))[?x]$

D. IT processing tracking and access control

In the last section we addressed hearsay tracking which is one aspect of provenance tracking. We did not address provenance tracking of tokens within IT systems. In our example token002 resides in knowledge repository B. It is based on token001, which resides in knowledge repository A. If an analyst queries the system that has access to the knowledge repositories A and B, then the information of token002 has to be ignored, since it provides no independent confirmation of the information. However, it might be the case that knowledge repository B but not repository A is available for queries; for example because of technical difficulties or because the agency of the analyst is not allowed to use repository A. In this case the system is supposed to use the information encoded in token002.

To support this functionality, for example, the query Que1 would have to be rephrased in the following way: *Find sequences s of tokens that support the proposition that Al Qaeda owns WMD, which meets the following additional requirement: there are no tokens y, z such that: (a) y resides in a repository that is available, (b) z is an element of the sequence s, (c) z is a copy of y, and (d) the sequence that is the result of replacing all occurrences of z in s by occurrences of y supports the proposition.*

Access control adds another layer of complexity. The query changes from 'Find me a sequence of tokens that support X' to 'Find me a sequence of tokens *that the user has access to* that support X'. Whether a user has access to a given token is determined by its security level and its security departments. We provide a detailed analysis of how to represent process tracking and access control in [1].

As we have seen in the case of Que2, the treatment of information about provenance on the same level as any other information enables queries that otherwise would not be possible. So far we looked at use cases that provided information for end-users. However, it is also useful for system administrators. For example, Que3 represents the query: *Find all secret records within repository B that are based on top secret records of repository A.*

Que3 $\mathit{Record}(?x)\ \&$
$\mathit{ResidesIn}(?x) = \mathit{repository\_B}\ \&$
$\mathit{ClassifiedAs}(?x\ \mathit{secret})\ \&$
$\exists y(\mathit{BasedOn}(?x\ y)\ \&$
$\mathit{ResidesIn}(y) = \mathit{repository\_A}\ \&$
$\mathit{ClassifiedAs}(y\ \mathit{top\_secret}))$

Tok4

Record(token005) &
ClassifiedAs(token005 secret) &
PropositionalContent(token005) =
(that (∃x (QueryAct(x) &
AskedBy(x analyst1234) &
Date(x) = 12.11.2011 &
PropositionalContent(x) = (that(
∃y (Owns (alQaeda y) & NuclWeap(y)))))

Another use case is to track which analyst accesses which data from which system. For example, assume that an analyst asks the query whether Al Qaeda owns nuclear weapons. At the same time the system answers the query, the system could generate Tok4, which records that the analyst asked a query, its date, as well as its propositional content. This information can be used to monitor who accesses which data from what sources and on what security level. It is also enables systems administrators to recognize if two independently working analysts are interested in the same content.

E. Non-propositional information

As mentioned above, the main focus of this paper is propositional information. However, it seems that our approach of treating tokens as first-class citizens in the domain of quantification could work not only for records but also for other tokens, for example pictures and audio files. Here is an example how one could represent the information about a picture that shows Maulana Masood Azhar visiting Baba Saab in Kandahar.

Tok5

Picture(token006) &
ResidesIn(token006) = repository_A &
ClassifiedAs (token006 top_secret) &
CreatedBy(token006) = agent1234 &
Source(token006) = source007 &
LocationDepicted(token006 babaSaab) &
About(token006 MasoodAzhar) &
About(token006 Shrine)

Security classification, the location of the token, and other 'meta-information' are provided in the same way as in previous examples. The main difference is that pictures do not encode propositional content. To tag the picture with keywords we are using the 'About' relationship (and subtypes of About like LocationDepicted). Since IKL lacks a syntactical distinction between predicates and individual constants, the second argument of the About-relationship can be filled by an individual (e.g, Masood Azhar) or a type (e.g., shrine). We are planning to further investigate the potential of our approach for the representation of non-propositional information in the future.

F. Treatment as a labeled deductive system

We note briefly here that the logic we have described may be considered a type of *labeled deductive system* [4]. The concept of a labeled deductive system is a generalization of the traditional notion of a logical system in which the consequence relation is defined relative to a system (algebra) of *labels* that modulate consequences that may be drawn in such systems. These systems have the advantage of incorporating what are traditionally viewed as meta-logical concepts (e.g. the rules for creating a proof) into the object language. Not only does this make, in many cases, for a more elegant description of the logic under consideration, it provides a unified description for logics that seem dissimilar on the surface but are in fact quite similar in terms of their underlying behavior.

The logic we describe can be considered a labeled deductive system for three reasons. First, the (sequences of) sentence tokens act as the labels of the system. Second, the operations on the labels encoded in the axioms for the SupportedBy relation define an algebra over the labels. Finally, the entailment relation for the system depends both on the content as well as the labels.

In fact, the application of labeled deduction to access control was proposed by Obrst and Nichols in [10], wherein they suggest (but did not develop) a labeled deductive system that operates over security labels in defining the consequence relation. Their proposal differs from our approach in two major ways. Our system quantifies over tokens, and enables multiple tokens of the same type to co-exist. Further, we pay specific attention to speech acts that assert tokens into a information system.

V. Discussion and future work

In this paper, we presented an ontologically-motivated approach to multi-level access control and provenance for information systems. We extended our previous work by widening the scope of the analysis to different use cases, most importantly hearsay provenance. Critical to our analysis is the role of linguistic tokens as the fundamental bearers of information and as the only entities capable of playing the causal role required to enforce access controls and track provenance within IT systems. These linguistic tokens might be bearers of information about speech acts with propositional content; these are used to enable hearsay provenance. We offered a formalized example of reasoning with provenance under multi-level access control. While the presentation was limited to access control and provenance in systems using overt logical reasoning processes, we would argue that the approach is applicable generally to information systems of all kinds (e.g., relational database systems or web-services).

In the future we are planning to extend this work to a theory of access control and provenance for non-overtly linguistic information bearing objects, such as audio, images, or video, and to account for effects of intentional degradation of information for "write-down" releases of information.

Acknowledgements

We thank Leo Obrst for his helpful comments and suggestions.

REFERENCES

[1] B. Andersen, F. Neuhaus. An Ontological Approach to Information Access Control and Provenance. In P. Costa, K. Laskey, L. Obrst (eds.): *Proceedings of the 2009 International Conference on Ontologies for the Intelligence Community* Fairfax, VA, USA, October 21-22, 2009. http://CEUR-WS.org/Vol-555/paper7.pdf.

[2] J.L. Austin. How To Do Things With Words 2nd Ed. Harvard University Press, Cambridge, 1975.

[3] D.E. Bell. Looking Back at the Bell-La Padula Model In Proceedings, *Annual Computer Security Applications Conference*, Tucson, 2005.

[4] D. Gabbay. Labelled Deductive Systems; Principles and Applications. Vol 1: Introduction. Oxford University Press, 1996

[5] P. Hayes. IKL Guide. http://www.ihmc.us/users/phayes/IKL/GUIDE/GUIDE.html

[6] P. Hayes, C. Menzel. IKL Specification Document. http://www.ihmc.us/users/phayes/IKL/SPEC/SPEC.html

[7] ISO/IEC 24707. Information technology – Common Logic (CL): a framework for a family of logic-based languages.

[8] F. Neuhaus, B. Andersen. The Bigger Picture – Speech Acts in Interaction with Ontology-based Information Systems. In M. Okada, B. Smith (eds): *Interdisciplinary Ontology* Vol. 2 (Proceedings of the Second Interdisciplinary Ontology Meeting), 2009, 45-56.

[9] United States Office of the Director of National Intelligence. Intelligence Community Directive Number 501. January, 2009.

[10] L. Obrst, D. Nichols. Context and Ontologies: Contextual Indexing of Ontological Expressions. Poster: AAAI 2005 Workshop on Context and Ontologies. AAAI, Pittsburgh, PA, 2005.

[11] J. Searle. Speech acts: An Essay in the Philosophy of Language. Cambridge University Press, New York, 1970.

[12] C.E. Shannon, W. Weaver. The Mathematical Theory of Communication. Urbana, Ill.: University of Illinois Press, 1975.

[13] L. Wetzel. Types and Tokens. The Stanford Encyclopedia of Philosophy (Winter 2008 Edition), Edward N. Zalta (ed.), URL = http://plato.stanford.edu/archives/win2008/entries/types-tokens/