An ontological approach to information access control and provenance

Bill Andersen
Ontology Works, Inc.
3600 O'Donnell Street
Suite 600
Baltimore, MD 21224

Fabian Neuhaus
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250

I. Introduction

Recently, the US Intelligence Community (IC) has elevated the need to share information from a slogan to a policy in the form of Intelligence Community Directive (ICD) 501.[1] The content of this directive can be summed up in one phrase: *responsibility to share*. However, the presence since the early 70s of well-established techniques and theory for multi-level security and access control have not solved the sharing problem; if they had, we would have been surprised by the advent of ICD 501. Of course access control is necessary – data stewards in member agencies will not share if, by doing so, they incur a risk of unauthorized release of information, especially when that information is related to specific sources and methods. Likewise, data consumers will not be incentivized to demand data held by stewards if that data cannot be trusted to be accurate, is of unknown origin, or lacks context. In other words, without providers, there is nothing to be shared and without consumers there is no reason to share. Thus, any viable policy for information sharing must rest on two interdependent notions: *access control* and *provenance*.

One could object that information sharing happens successfully all the time, e.g. on the internet via search engines. The conduct of science provides an example, where theories and experimental results are freely exchanged in an open, peer-reviewed environment. The key term here, however, is *open*. Information sharing in the intelligence context shares few of the features of information sharing in science, save the universal desire to produce and use information. The community with interest in sharing intelligence information is, unlike science, *closed* and the information itself is, like science, *epistemically unstable*. It is these features of information sharing in the intelligence setting that underscore the need for both access control and provenance.

In an attempt to reconcile these two problems in a unified way, we take a step back and examine them through an ontological lens. That is, rather that starting from an abstract mathematical framework (e.g. [1]) we begin our investigation anew by asking what kinds of objects are involved and what are their properties.

A. Example

In the following we will discuss our approach with the help of the following scenario. Assume that a US agent reports a sighting of Osama bin Laden in Kandahar. 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 revealed. The second agency stores the information about Osama's location within their knowledge repository B and classify it as secret. Assume further, that in the same time frame the New York Times reports that Osama is in Kandahar. 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 Osama is in Kandahar. However, it is classified differently in the knowledge repositories A, B, and C (as secret, top secret, and unclassified, respectively). To further muddy the water, assume that repository B also contains a report from an agent whose notoriously unreliable contact claims that Osama is in Somalia. Thus, repository B contains conflicting information about Osama's location.[2]

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 Osama is in Afghanistan.* For the sake of simplicity, let's further assume that the knowledge repositories contain no other entries about Osama's whereabouts than the ones mentioned above. The correct answer of 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 New York Times report in repository C. The fact that the information provided by the US agent in Kandahar is classified should not prevent the analyst to access the information based on news reports, although in some sense it is the *same* information. If the analyst has access to top secret information, the system should provide two: the New York Times report and the original record by the US agent

[1] http://www.dni.gov/electronic_ reading_ room/ICD_ 501.pdf

[2] For the sake of simplicity, we ignore do not treat the role of time explicitly and just assume that the statements about Osama's location are valid during the same time period.

in repository A. The system should not provide the record in knowledge base B, since it is merely a copy of the one in repository A.

The rest of the papers is as follows. Section II presents an ontological analysis of the entities relevant to access control and provenance. In Section III we a worked formalized example, based on the above ontology, of reasoning over information under provenance and access control. We conclude with discussion and directions for future work in Section IV.

II. Ontology of access control

Our approach to a theory of access control is ontological rather than procedural.[3] By first examining and fixing the relevant kinds of entities involved in access control, we hope to provide a firm foundation for an evolving formal theory that addresses access control and provenance.

Central to any discussion is of course the what is meant by *information* itself. Our account is tailored to the case of information systems dealing with propositional information encoded in formal language expressions. In this paper we will explicitly ignore information encoded in images, video, audio and other like forms of common digitally encoded media. We believe the approach taken here can be extended to those other kinds of infomration-bearing entities and intend to do so in future work.

A. Information

The notion of information we adopt is that from Shannon ([7]). On this view, information is an abstract notion meaning conventionally that it is non-spatio-temporal in nature. It is hard to imagine how we might control access to or provenance of anything abstract, let alone abstract objects that carry information, save through some sort of physical encoding. Thus, the objects of access control, whatever they are, must participate in the causal structure of information systems. Whatever we mean by information in this context, it must be analyzable in terms of concrete spatio-temporal objects manipulable by computer systems. The reason for this is that computational mechanisms have causal influence only over objects that are ultimately encoded as patterns of electrons or some other physical mechanism.

B. Formal languages and sentences

The kind of information that we are concerned with in this paper is that which can be encoded in sentences of a formal language. We take the notion of a formal language as primitive without further analysis. In the discussion that follows, we assume well-formedness of any expressions generated from such grammars. Dealing with ill-formed expressions is beyond the scope of this paper. We also assume a truth-functional semantics for each language such that to well-formed expressions (sentences) of the language it is possible to assign a truth value.

The fundamental unit of information we consider is that of the *sentence*. By sentence we mean a well-formed expression of a language that encodes a proposition, contains no free variables, and is primitive in that it cannot be further decomposed into parts that are themselves truth-bearing. Note that this notion of sentence permits closed quantified formulae that contain connectives but not, for example, a ground formula with connectives.

Note also that this encoding precludes two kinds of encoding devices in common use. First is the use of so-called "denormalized" forms in relational models wherein a single record could be decomposed (but is not, by design) into several distinct sentences. Second is the use of names which themselves encode structured information. Both devices are used commonly in information systems for efficiency's sake or to reduce storage requirements.

C. Type and token

Our usage of 'sentence' in the previous section has been ambiguous. We need to distinguish between sentence types and sentence tokens [8]. To illustrate the distinction, please count the number of sentences below:

The cat is on the mat.
$[*]$ Die Katze ist auf der Matte.
The cat is on the mat.

If you counted three, then you were counting sentence tokens. If you counted two (one German sentence and one English), then you were counting sentence types. If you counted just one, you were counting the number of propositions encoded by the sentence tokens.

Sentence types as well as propositions are conventionaly taken to be abstract entities and thus cannot be objects of access control and provenance for the reasons given above. In contrast, sentence tokens are physical entities (e.g., the distribution of ink on a sheet of paper or arrangements of electric charges in a chip). Different sentence tokens of the same types might have different properties. E.g., the three sentence tokens in $[*]$ are distinguished by their physical locations. More importantly, different sentence tokens of the same type can differ with respect to their security properties: a IT system might contain an classified encoding of a proposition P and an unclassified encoding of P. The latter encoding requires no protection whereas the former does. This would not be possible if access control would apply to sentence types. In this paper we consider sentence tokens as the primary bearers of security properties and access control.[4]

D. Copying, recoding, and synthesis

On a token-based view of access control, 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 events (copying, synthesis, and recoding) that make new tokens from old ones. Depending

[3] The Bell-La Padula security model is one example where secure states of a system are defined by a state machine model ([1]).

[4] In a previous paper we have used speech acts to fulfill this role.[6]

on the type of operation involved, properties relevant to access control will need to be preserved.

To invoke some paper examples to avoid the complication of the inner workings of computer systems, suppose there is a document A that is classified as top secret. Simply copying the document on a photocopier will not declassify the resulting copy – its security protection properties must be preserved by copying. Likewise, a recoding of the document into a document in another language (say German) should not render it unclassified. In addition, a document that is the synthesis of protected documents, e.g. a summary report, needs to be protected as well.

One important case of synthesis is that of automated logical reasoning. Computational inference procedures operate on stored tokens to produce other tokens. The level of protection for these newly created tokens depends on the level of protection of the original tokens. The system must behave as if, associated with each of these procedures for token synthesis, there is a function from the protection properties of the process inputs to the process outputs. For example, if we have a rule of our proof calculus $A; B \vdash (A\&B)$ then the system needs to be able to determine the access control properties of the token that encodes $(A\&B)$ based on the access control properties of the source tokens encoding A and B.

E. Systems and boundaries

In all the discussion above, we have been somewhat loose in using the term *system* without defining it. By system we mean a physical object that is capable of accepting information encoded in some appropriate language and of accepting and responding to queries posed in some appropriate language with the result being a *release* of tokens encoding the query response. One example of such a system would be a relational database management system.

This behavior is necessary but not sufficient – we require further that the system act as a kind of boundary that respects whatever calculus we require for access controls on released information. That is, the system must allow access to stored information only through specified processes and through no other means.

We assume further that systems are under the control of agents responsible for their operation. While not essential for the formal treatment presented here, any practical application will require such stewardship relations be taken into account as security and sharing policies makes sense only in the enviroment of a contract between such agents.

F. Security classification

Finally, since we will be discussing the use of security classifications in what follows, we should say a few words about what those are. A security labeling system in this context will consist 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-Quaeda* would be more specific than the compartment *terrorist group*). Ontologically speaking, security levels and compartments are properties – social artifacts that are dependent upon a community of agents that mutually agrees to the storage and access of information using the labeling system.

III. Formalization of access control

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.

While the techniques here may be extended to arbitrary types of information systems, in this paper we are mainly concerned with presenting the framework that would allow a logic-based system to achieve this goal. For this reason we assume that the reasoner supports a very expressive language, at least as expressive as IKL extended by two athletic modality operators $\Diamond$ and $\Box$.[5] IKL is an extension of CLIF which itself is the interchange format of the ISO standard Common Logic. 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 $s, s1, s2$ as sequence variables – variables that range over finite sequences of objects.)

Let's return to the example from the the introduction. The knowledge repository A of the first agency contains a token that expresses *Osama is in Kandahar*. Further, the repository contains some 'metainformation' about this entry; for example, the information is top secret, and the source of the information was the agent with the identifier 1234. We suggest to represent this content in the following way:

Tok1
Record(token001) &
ResidesIn(token001) = repository_A &
PropositionalContent(token001) =
(that (LocatedIn (osama kandahar))) &
ClassifiedAs (token001 top_secret) &
Compartment (token001 al-quaeda_cmpt) &
Compartment (token001 afghanistan_cmpt) &
Source(token001) = agent1234

The name 'token001' is a name of the record that resides in the repository A. The function ResidesIn represents token001's containment in A. The next formula expresses the propositional content that is encoded in the token.[6] In the next lines represent the 'metadata': it's classification, compartments

[5] The details of the extension of IKL by modal operators are beyond the scope of this paper. We assume that the propositional fragment of the language satisfy the S5 axioms.

[6] We mentioned in Section II that both sentences and propositions were abstract and thus unsuitable for the causal role tokens play. Here 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. Another formalization could be carried out with quotation in a meta-language.

and the source. Additional information, e.g., the time of the creation of the record could be added.

In our example, the information about Osama's whereabouts is shared with another agency, and stored in the knowledge repository B. This is represented in the following way:

Tok2
Record(token002) &
ResidesIn(token002) = repository_B &
PropositionalContent(token002) =
(that (LocatedIn (osama kandahar))) &
ClassifiedAs(token002 secret) &
CopyOf(token002, token001) &
ResidesIn(token001) = repository_A &
Source(token002) = nytimes

Notice, that token001 and token002 are both records that encode the same propositional content. The main differences between them is that they reside in different repositories and that token002 has a different access restrictions. Further, the knowledge repository B does not contain any information about the source of the information. The fact that the entry in knowledge base B is originated from repository A is expressed explicitly by asserting that token002 is a copy of token001 and that token001 resides in the repository A. The 'CopyOf' relation abstracts away the event of copying, thus recording part of token002's causal history. This will enable a reasoner to detect that token001 and token002 do not provide independent information about the location of Osama.

Since we will need it in the next section we also include the last part of our example: the information from the New York Times article that is stored in knowledge repository C.

Tok3
Record(token003) &
ResidesIn(token003) = repository_C &
PropositionalContent(token003) =
(that (LocatedIn (osama kandahar))) &
ClassifiedAs(token003 unclassified)

B. The support relationship

We now add another relationship "SupportedBy" between a proposition and zero or more records. The goal of this relation is to capture not only the propositional content that is captured one record, but what is logically entailed by a set of these records. One problem we need to address is that in the framework of a classical logic a contradictory information logically entail any proposition. Assume we have an ontology-based information system with a classical reasoner and access to spatial information sufficiently strong to prove that Osama cannot be located in Kandahar and (at the same time) be in Somalia. In our example, one agent claims that Osama is in Kandahar and the other agent claims that Osama is in Somalia. If we were provide both information in an unaugmented way to this system, the reasoner would 'use' the logical contradiction to prove any query – and thus the IT system would become useless. Our goal is to enable to limited reasoning with contradictory information, but to prevent the system from 'exploding'.[7] This is achieved with the help of the athletic modality operators; $\Diamond$ is read as 'it is possible' and $\Box$ is read as 'it is necessary'.

Instead of SupportedBy(that(A), s) we write also 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.

Ax1 *Record(x) → SupportedBy(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 $(A[s]\&\Diamond A\&\Box(A \rightarrow B)) \rightarrow B[s]$

Ax1 expresses that every record supports its (own) propositional content. Further, ever 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 record 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 impossible that (A & B) can be true; thus in this case A[s_1] & B[s_2] do not imply (A & B)[[s_1, s_2]. Ax4 expresses the following: if the sequence s supports a proposition A, and the proposition is possibly true, and A necessarily implies B, then the sequence s also supports the proposition B. The axiom ensures that a sequence of record does not only support a conjunction of their propositional contents but also the logical consequences of the propositions – provided that the records do not support logically inconsistent propositions. The reason for the latter constraint is that without it a sequence of assertions of contradicting information would support every proposition because, as discussed above, in classical logic a logically false formula will entail any formula.

C. Reasoning with SupportedBy

The support relationship is used to enable queries for information that supports a given hypothesis. In the rest of this section we will show how that works with the help of the running example from the introduction. First, we consider one example where we ignore the role of security classification and the fact that token tok2 is a copy of tok1. Later we will discuss how these more complicated examples are treated.

Let's assume that the system has access to an ontology that either contains or logically entails the following background information: If somebody is located in Kandahar, then he is located in Afghanistan (Bgnd1).

[7] This feature of classical logic is well-known by logicians and is one of the main driving forces behind the study of relevance logic and paraconsistent logics . Since we are defining a relationship between types and propositions our goal is different than the one in relevance logic or paraconsistent logic, but our "SupportedBy" could be shortly characterized as a non-monotonic variant of the strict implication.

Bgnd1 *LocatedIn(x kandahar) → LocatedIn(x afghanistan)*

In our example, the analysts query the system for information that support the hypothesis that Osama is in Afghanistan. For starters, we can represent the query 'Find all sequences of records that are supporting the proposition Osama is located in Afghanistan' in the following way:

Que1 *LocatedIn (osama afghanistan)[?s]*

Note that IKL itself does not provide any convention how to express queries, we use question marks in front of variables to mark the variables that are supposed to be bound by reasoner.

When an analyst enters the query qu1 into the system, it tries to find a sequence of tokens that enables it to support the claim that Osama is located in Afghanistan. For example, the system would try to proof that token001 supports this proposition. The proof is, in fact, very straight forward:

Proof: LocatedIn (osama afghanistan) [token001]

1) LocatedIn(x kandahar) → LocatedIn(x afghanistan)[ ]
2) LocatedIn (osama kandahar) [token001]
3) $\Diamond$ (∀x (LocatedIn(x kandahar) → LocatedIn(x afghanistan))
   & LocatedIn (osama kandahar))
4) (∀x (LocatedIn(x kandahar) → LocatedIn(x afghanistan))
   & LocatedIn (osama kandahar)) [token001]
5) $\Box$((∀x (LocatedIn(x kandahar) → LocatedIn(x afghanistan))
   & LocatedIn (osama kandahar))
   → LocatedIn(osama afghanistan))
6) LocatedIn (osama afghanistan) [token001]

Line 1 of the proof is an immediate consequence of Bgnd1 and Ax2. Line 2 follows from Tok1 and Ax1. Line 3 and 5 are theorems of the modal logic we are assuming. The first three lines and Ax3 entail line 4 of the proof. Line 4, 3, and 5 together with Ax4 entail line 6. Q.E.D.[8]

While this proof is admittedly very simple, it is sufficient to show how proofs in general work: the 'background information' that is provided to the system as truths (like Kandahar is part of Afghanistan), lead to support-statements with an empty sequence by axiom Ax2. Formulas that express the content of records in knowledge repository (like Tok1) lead to supports-statements via axiom Ax1 that contain lists with only one element. In our example we these axioms only once each, but in more complex examples one would have to use these axioms repeatedly. The resulting supports-statements can be combined to more complex ones with the help of axioms Ax3. In our example, the proposition is only supported by one token, but it could be an arbitrary long list of tokens. For example, lets assume the system has access to a record (token005) in a knowledge repository according to which Osama is only in Kandahar if he is sick. Based on previous information it would be able to prove:

Sick(osama)[token001 token005]

D. Redundant answers and copies

We continue with our discussion of the query in our example (*Find records that support the hypothesis that Osama is in Afghanistan*). In the previous section we a made a number of simplifications. 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 as sketched above 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 qu1, in spite of the fact that token001 supports the proposition on its own and token005 does not contribute anything. For the sake of brevity we will not further explore these kind of redundancies; it is quite easy to avoid them.

More interestingly, we need to address the problem that information is shared between agencies and thus we need to detect whether an entry in a knowledge repository is an original or just a copy. In the following we will write $x \in s$ in order to express that x is one of the components of the sequence s; e.g., $b \in (a\ b\ c)$. Further, we will write s^x_y to denote the sequence that is the result of substituting all occurrences of x by occurrences of y; e.g., if $s = (a\ b\ c)$, then $s^b_d = (a\ d\ c)$.

Let's consider the intended behavior using our example: token002 is a copy of token001 that 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 about Osama's whereabouts. 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.

In order to achieve this behavior, query qu1 is replaced qu2 that can be read as follows: find sequences s of tokens that support the proposition that Osama is in afghanistan that meets the following additional requirement: there are no tokens x, y such that: (a) x resides in a repository that is available, (b) y is an element of the sequence s, (c) y is a copy of x, and (d) the sequence that is the result of replacing all occurrences of y in s by occurrences of x supports the proposition:

Que2

LocatedIn (osama afghanistan)$[?s]$ &
$\sim\exists x \exists y$
(Available(ResidesIn(x)) &
$(y \in ?s)$ &
CopyOf(y, x) &
LocatedIn (osama afghanistan)$[?s^y_x])$

Bgnd2 *Available(repository_A)*

[8] One can prove that the proposition is supported by token002 and token003 in exactly the same way.

This query effectively ignores tokens that are copies of other tokens that reside in available repositories. Let's assume Bgnd2. In this case query qu1 finds three solutions: token001, token002, and token002; in contrast , query qu2 does only find two solutions, namely token001 and token003.

E. Access control

So far we have discussed queries where an analyst looks for information that support a given hypothesis. However, so far we have not taken into consideration that different analyst have access to different kind of information. Thus, the question changes to: Is the hypothesis supported by tokens that the user has access to?

Access control is provided by security classifications and compartments. In our example we use the classification system commonly in use by the US IC; however, any classification system would do. To enable reasoning about access control we use a strict ordering relation $<$ on the set of security levels and compartments. Some examples for compartments are included in Bgnd3. (We use the suffix '_cmpt ' in order to avoid ambiguities; e.g. one should not confuse the country of Afghanistan with the compartment in a access control system.)

Bgnd3

unclassified $<$ *confidential*
confidential $<$ *secret*
secret $<$ *top_secret*
afghanistan_cmpt $<$ *asia_cmpt*
asia_cmpt $<$ *world_cmpt*
al-quaeda_cmpt $<$ *terrorism_cmpt*
$<$ *is transitive.*
$<$ *is asymmetric.*

In addition we need to introduce ClearedFor, a relationship that holds between users and either security levels or compartments. The only relevant axiom for ClearedFor is that if some user is cleared for a security level (compartment), then this implies a clearance for any security level (compartment) lower in the hierarchy. For example, if an analyst is cleared for top secret documents concerning terrorism worldwide, then she will have access to a secret document with the compartments asia and terrorism.

Ax5 $\mathit{ClearedFor}(x\ y)\&(z < y) \rightarrow \mathit{ClearedFor}(x\ z)$

Based on these axioms we now define *proposition A is supported by sequence s with respect to user x* which we abbreviate as $A[s]_x$:

$$A[s]_x =_{df} A[s]\& \\ \forall y \forall z(((y \in s)\&\mathit{ClassifiedAs}(y\ z)) \rightarrow \mathit{ClearedFor}(x\ z))\& \\ \forall y \forall z(((y \in s)\&\mathit{Compartment}(y\ z)) \rightarrow \mathit{ClearedFor}(x\ z))$$

With the help of this definition a query by an analyst that wants to check whether Osama is in Afghanistan could be represented as follows (we skip for the moment the result of last section):

Que3 *LocatedIn (osama afghanistan)*$[?s]_{\mathit{analyst001}}$ *&*

Let's assume that the system has the following information about this user:

User1

name(analyst001) = 'Nathan Hale'
ClearedFor(analyst001 top_secret)
ClearedFor(analyst001 asia_cmpt)
ClearedFor(analyst001 terrorism_cmpt)

From User1, axiom Ax5, and Bgnd3 it follows that analyst001 is cleared to access top-secret records about Al-Quaeda and Afghanistan. Thus, analyst001 is allowed to access token001, which, as discussed before, supports the proposition that Osama is in Afghanistan. This is true for token002 and token003 as well. Thus, the analyst would get all three responses back. If analyst001 would be cleared only for secure (and lower) or would not be cleared for the Afghanistan-comparment, then analyst001 would have access to token002 and token003 but not token001.

IV. Discussion and future work

In this paper, we presented an ontologically-motivated approach to multi-level access control and provenance for information systems. In it we recognize 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. Based on the rump ontology presented, 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 argue the approach is applicable generally to information systems of all kinds (e.g. relational database systems or web-services).

We would like 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.

References

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

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

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

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

[5] F. Neuhaus., Ontology-based technologies – Technology transfer from bioinformatics. In K. B. Laskey, D. Wijesekera (eds): *Ontology for the Intelligence Community (OIC-2008) – Towards Effective Exploitation and Integration of Intelligence Resources*, 2009.

[6] 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.

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

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