Toward an Architectural Framework to Improve Accountability in the Use of Electronic Records

G.E. LYON[1], A. MINK[2], and R. E. VAN DYCK[2]

Abstract—Sensitive electronic record systems (*ERS*s) raise questions about their proper use. *Insider-threat* involves hidden, unknown and unanticipated activities that constitute unacceptable use of an ERS, even while operating within individual access privileges. Insider-threat detection and control is an ERS monitoring and management challenge of the first order. A flexible preliminary framework can encourage discussion and comparison among various monitoring elements for the insider-threat. Responding to a lack of such a framework, one is sketched here: It employs two perspectives of an ERS user—structural and intentional. The structural view is short term, whereas the intentional view seeks to discover general content topics of interest to a user, and to follow these over time. Discussion includes details of a possible architecture that uses untrained classification methods to amplify the concern set beyond that specifically defined at the onset of monitoring. The general framework may expedite development of common guidelines and methodologies to monitor insider threats. Although developed for medical services (e.g., an E-Health RS), the framework likely has applicability in other similar database areas such as security and intelligence archiving.

Index Terms—Architecture, digital records, electronic record systems, framework, insider-threat, monitoring

I. Background

Increasingly*, archival records* must be *electronic* to meet challenges of economy, accuracy, volume and speed; these are demanding requirements that only information technology (*IT*) can satisfy. Many application areas share electronic record systems requirements. For example, in many respects circumstances driving the use of ERSs (*electronic record systems*) in the medical industry parallel conditions in the intelligence community. Both fields generate vast archives of shared but confidential information. Each has a need for rapid dissemination and sharing while nonetheless maintaining a level of accountability for each user.

Accounting for proper ERS use is a genuine challenge. Assuming ERS users adhere to authorized access in a narrow, transactional sense of IT security does not guarantee the propriety of longitudinal, or long-term, accesses. Generally local logs (or log summaries) exist of database accesses and requests. But these logs become voluminous and only provide raw, difficult to interpret and uncorrelated information. To be more effective, these transaction logs should be correlated automatically against affiliations and roles of users. The roles themselves may be poorly defined and need refinement (one important task of insider-threat monitoring). An example might be a senior doctor in a hospital who runs many summary statistics. Suppose closer examination reveals that this physician, who is not an administrator, owns a local retail pharmacy. Fresh hospital statistics provide a more efficient inventory for his business. This is a definite competitive advantage. In this case, accountability oversight has identified a defined hospital role that has overly broad privileges. A similar oversight requirement can apply to other users of electronically held sensitive materials. For example, an analyst in the intelligence community, who is not a supervisor, has submitted a report to a database and in that process is browsing through all other reports in the database.

II. Architectural Framework

Our objective is to explore an architectural framework that accommodates mechanisms of use and accountability for long-term supervision of ERS database use. The attempt here is designed to provoke and sustain discussion on the propriety of details in EHR monitoring. Such discussion can only help researchers in the field develop a common frame of discourse against which progress can be measured. A framework—and its evolutions to correct deficiencies--will also begin to identify clearly those areas in which substantial technical challenges lie. Figure 1 (see below) depicts the general setting. Blue designates the materials of interest and their attributions, which could be keywords, word counts, or other features, such as those of photographs (values, hues, saturations, etc.).

Some Assumptions.

To simplify discussion, we have assumed that the attributed database materials (blue colored section in Figure 1) do exist. In practice, getting documents usefully attributed may in itself pose a substantial challenge. We do not discuss the problems of automated document tagging here.

This work was supported in part by the National Security Agency, Advanced Research and Development Activity (ARDA)

Authors are with the National Institute of Standards and Technology, Gaithersburg, MD 20899-8951 USA

Mention of commercial products or services is for illustrative purposes only—it does not constitute any endorsement, express or implied.

[1] Division 897, Software Diagnostics & Conformance Testing (lyon@nist.gov)
[2] Division 892, Advanced Network Technologies (amink@nist.gov; Robert.vanDyck@nist.gov)

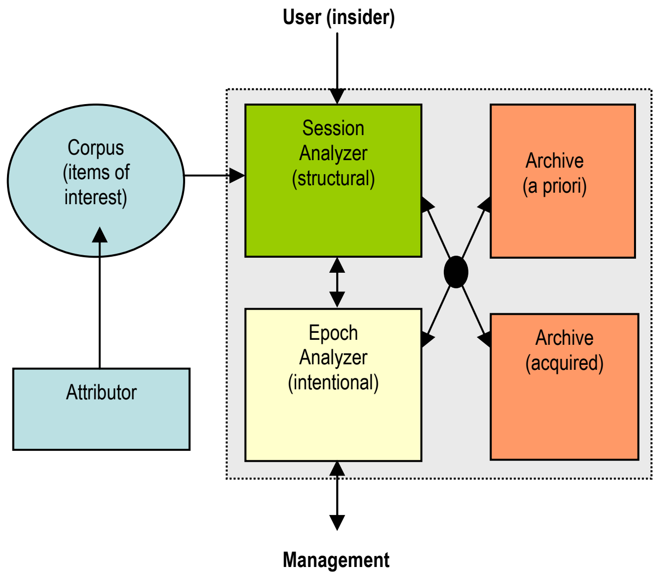

User (insider)
Corpus (items of interest)
Session Analyzer (structural)
Archive (a priori)
Epoch Analyzer (intentional)
Archive (acquired)
Attributor
Management

Figure 1. Context and General Architecture

The main *insider- threat monitoring* architecture is within dotted lines on Figure 1 (above). User requests, indices of found items and metadata from items actually retrieved feed into the upper left corner. A session analyzer monitors transactions within each retrieval session. Sessions are checked especially closely for odd patterns of retrieval. The epoch analyzer has longer-term concerns and is a separate component. Epochs are examined for interests that the user has displayed. These user's interests are compared against a cohort of similar colleagues over similar periods, and also, against concerns that management has about the database. In each phase of analysis—session or epoch—analysis is driven by internal data on the user. This data appears in Figure 1 as *a priori* or acquired archive storage (depicted in orange); a later section (*e.g.*, see orange parts of Figure 2) expands upon possible roles the data might serve. The principal objective here is to suggest some details within the dotted area of the general framework: Such details may provoke constructive disagreements that result in alternative formulations within the same context. This will be highly productive in comparing approaches to speed progress.

Building upon the General Framework.

A flexible preliminary framework can encourage discussion and comparison among various monitoring elements for the insider-threat. Imagine sketching out a "straw man" with more details than the general layout of Figure 1. One example of such an "insider-threat" discussion vehicle employs perspectives to address both "structural" and "intentional" views of an ERS user. Of the two, structural perspective seems more short term. Consider a pattern of searches conducted within a single retrieval session. Structural information exposes threats that might exist in certain questionable (often indirect) searching practices. An example would be using complimentary summation searches (e.g. payroll sum for [all engineers] and payroll sum for [all engineers - X] to discover forbidden individual values, such as individual X's salary. Short-term structural monitoring involves detecting formal search manipulations that seem out of place, overly elaborate, or nonsensical.

The intentional view, on the other hand, seeks to discover in content topics the actual, focused interests of a user. Users' search patterns as they explore topics and interests are—at best—secondary. However, knowing which *insider* interests predominate is *sine qua non*. Our straw man will follow user interests over time. Imagine a coarse intentional summary as being a weighted topic vector. From such vectors—which comprise the principal data architecture of our framework—user "interests" can be surmised and recorded (in further vectors) for later correlation against management concerns. The straw man ends with untrained[3] classification methods that amplify the set of concerns beyond those specifically stated at the onset of monitoring. Feedback from management and the classifier stages will, over time, modify the initial set of framework interests and concerns. This modification is likely to remain semi-manual for some time.

III. A Straw Man

We now develop now a useful straw man within the general architectural framework of Figure 1. Figure 2 (see next page) gives a component dependency view of this straw man. Component blocks on Figure 2 are referenced using their letter designation enclosed in parentheses, *e.g.* as (A), or (H). Related functions in Figures 1 and 2 are colored the same, *e.g.,* green denotes session structural analysis. The framework begins by monitoring a user's interaction with the ERS. There are—see component (A) in figure 2—queries Q, hits H and retrievals R in a recent time interval. This interaction drives a structural analyzer, (B), and an intentional analyzer, (E). Structural model (B) derives largely from interaction behavior that does not involve much use of metadata. The structural (session) component looks for patterns of access and is highly application dependent. It compares the *modus operandi* of a user against average behavior demonstrated at similar positions within the organization. Appropriateness of behavioral form is an issue and contrasts to checking actual content, which is the purview of the intentional stages. A keyword lookup (C) is a simple, small, largely lexical level that feeds both the structural model and the intentional model. It has fixed and dynamically defined elements. The metadata mapping (D) employs keyword lookup and embedded metadata. In a very comprehensive setting, this structural component may need terms mapped from one classification to another. For example, an ontological mapping of medical terminology might carry SNOMED-CT (110K terms) into ICD-10CM (140K terms). Such a "normalization" mapping is especially important when the database of materials is heterogeneous. Whenever

[3] Trained classification has seen input that is labeled correctly. With untrained classification, categories evolve but their content is not labeled.

materials lack a common origin, they will likely use some terms in different ways.

The structural model, interpreted at (B), should go beyond any straightforward checking of privileges. It uses rights and roles tables (in F) from the ERS's conventional system access process, augmented by its own rules as necessary (also in F). It looks for odd behaviors during a session and records these along with more obvious factors of document types, sources, etc. It begins the accumulation of various profiles of users (H). (Note specifically that this straw man, which is a shadow system, *does not* affect conventional system access rights of a user.) The intentional model starts with topic vectors (E) of weighted entries. These represent a short-term, normalized summary of user interactions and topics. Both the structural and intentional models contribute to (G) maintaining a user set of profiles (short, medium, and long) as well as (H) a dynamic repository for these profiles and summary statistics over different time scales. The analysis and visualization component of the intentional model (J) uses data mining, supervised and unsupervised learning methods. Examples might include decision trees, neural networks, support vector machines, and market basket analysis.

Context.

One of the challenges of insider-threat monitoring is that patterns of violation have not yet been explicitly expressed: management is unaware of whatever threatening practice is going on. The problem thus involves discovering patterns, usually over an extended period of time. Our straw man would use classifiers (statistical and pre-loaded) and clustering to find anomalies in database/user access patterns. Although others have proposed such anomaly detection methods—database literature contains much current thinking—our framework differs in two significant ways. First, we apply these concepts as a richer combination of both application and network layer data. Second, the straw man does not assume that training data (correctly labeled input) is available for the classification scheme.

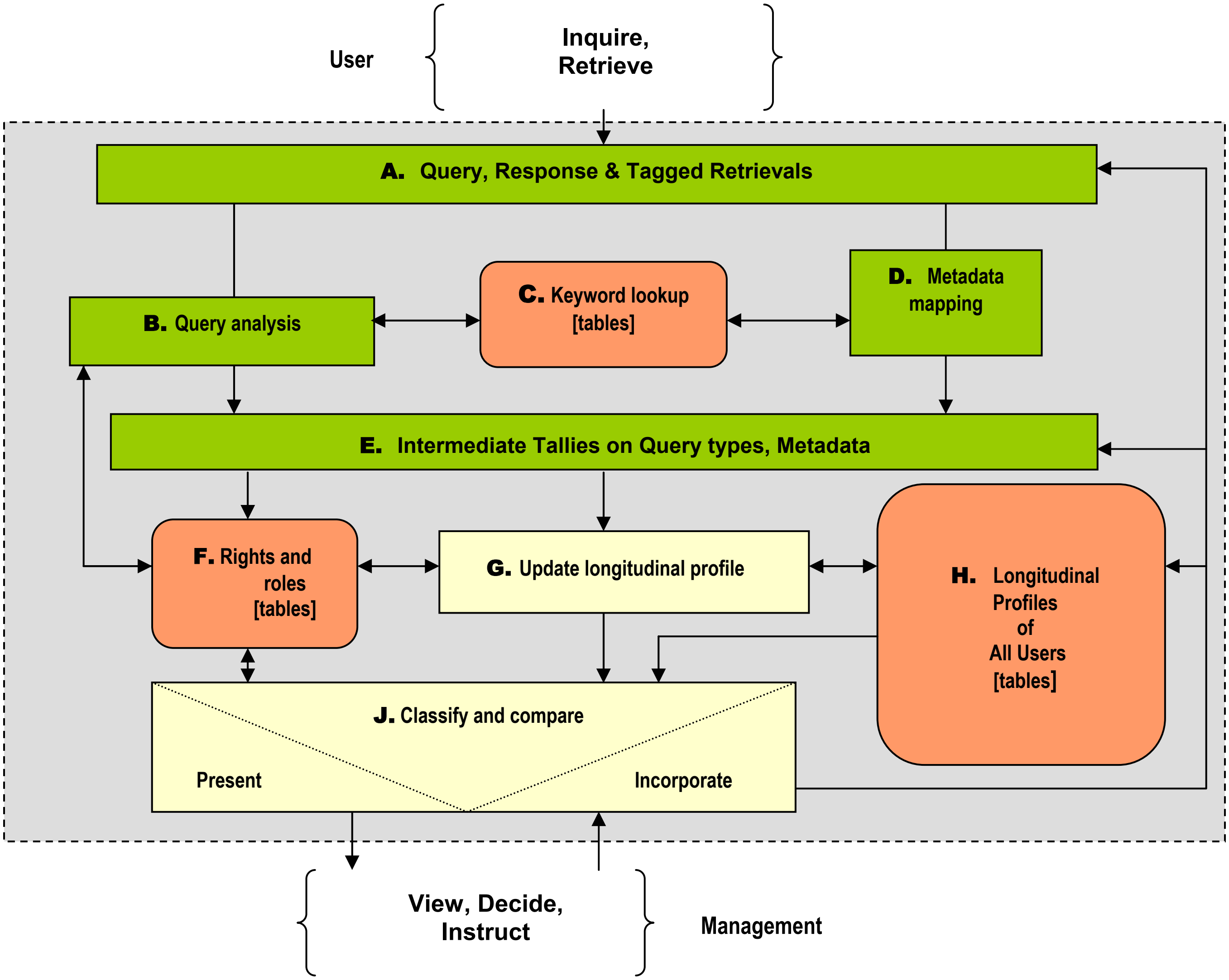

User
Inquire, Retrieve
A. Query, Response & Tagged Retrievals
B. Query analysis
C. Keyword lookup [tables]
D. Metadata mapping
E. Intermediate Tallies on Query types, Metadata
F. Rights and roles [tables]
G. Update longitudinal profile
H. Longitudinal Profiles of All Users [tables]
J. Classify and compare
Present
Incorporate
View, Decide, Instruct
Management

Figure 2. Straw Man for ERS Accountability

Data Space in the Straw Man.

Suppose user *D* wants to know about *"systemic TB" AND "Medicare patient."* Call this search pattern *P*. He initiates a search with pattern P, Q(P), on an information system of sensitive material. The information system server takes this request and examines its available archives. A number of hits {H} result and are conveyed—by title—to D. From these, D chooses a higher-interest subset {R} to retrieve. Pattern P, and sets {H} and {R} lie in a query space and represent a multitude of topics, in which "D" expresses *interests*, both intended and unintended. We can think of this *topic space* as a weighted list of individual categories that are revealed by specific query/hit/retrieval triples (Q(P), {H}, {R}). There is, however, a complement to these interests: Search Q(P) and the record sets {H} and {R} also relate to *concerns* held by management of the information system.

Neither user D's *interest* nor management's *concern* is necessarily explicit in D's inquiries and responses; a query on "Modern Penicillin Variants" and a retrieved report from the hit set on "Common Wasps and Bees" may indicate a latent interest in *anaphylactic shock, alternative treatments,* or, perhaps something entirely different. Any monitoring thus needs an intermediate phase that first maps queries into more common *topics*. These topics indicate—directly or indirectly—interests and concerns. Many superficial features will likely be lost in this mapping. The challenge is to identify hidden but more invariant interests and concerns. These truly matter. From a perspective of the host institution, D should be retrieving only sensitive materials appropriate to his roles and duties within the institution and should be seeking no others. An "interest" may actually be a suspect pattern of inquiry that has been detected in a session. Eventually, the host institution would like to analyze and interpret these suspect patterns to know that all users exhibit no indications of irresponsible use.

Query Space → Topic Space ←→ Concern Space

The diagram above, along with Figures 1 and 2, illustrates data paths within our proposed framework:

1. An overt *Query Space* is built upon immediately measurable triples, (Q(P), {H}, {R}), pattern rules and other immediately measurable factors (see Table 1, below)
2. An intermediate *Topic Space* is represented as an n-tuple, (T1, … , Tn) where Ti is the weight of Topic i
3. A latent *Interests and Concerns Space* represents roles, use and constraints.
4. Mappings among the three spaces as appropriate

These three spaces are explained in further detail in the following section.

• *Name*	• *Level*	• *Comments*
User ID	Application, system	Assume access authorization[4]
Document/ Patient ID	Application	Is unique from all others
Categories	Application	Subdivision(s) of the document or patient record (not unique, e.g., *cardiac*)
Location(s)	Network	Source and destination IP addresses, not physical locations. Assigned or DHCP
Time(s)	Application	
Activity Level	Application	
Query Patterns	Application	Is within session Flags especially inefficient or indirect searches as suspect
Network Info.	Network	Ports, Protocol
Action	Application	Read, copy, print, modify, *etc.*

Table 1. Application and network level data in each request.

[4] This study assumes conventional security works perfectly. Thus at the transaction level, any granted access is *bona fide*. Our question is whether, over time, the fuller picture of these authorized individual accesses remains proper.

IV. More Straw Man Details

Query Space.

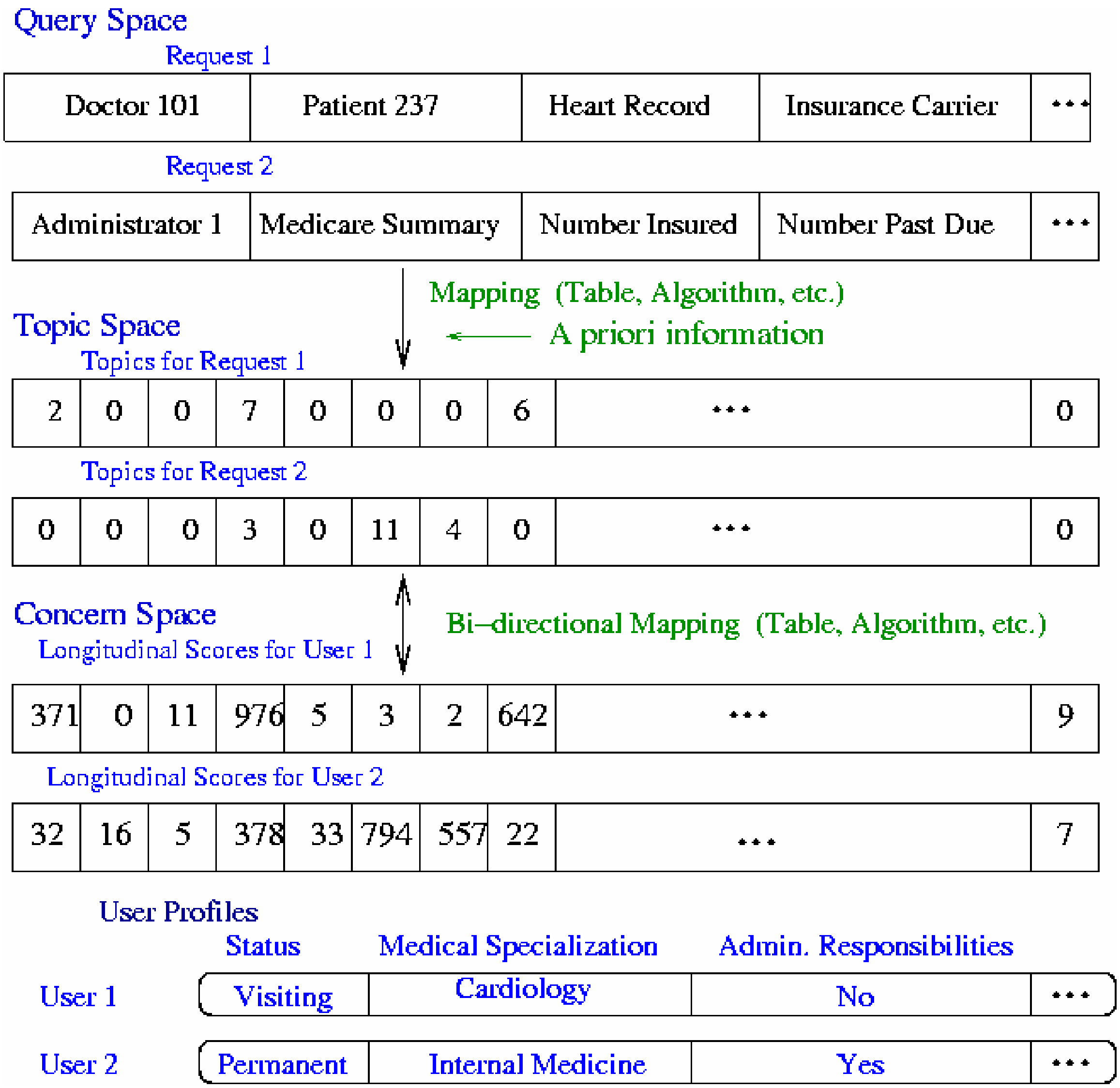
The query space establishes information that drives subsequent processing and analysis. This information includes database processing required to provide the user with the relevant data, as well as our added factors. Let a user's query on pattern P be Q(P) and the information headings and titles revealed be {H}. These, plus documents retrieved, {R}, largely define tangible events with which we work. Application-level transactions are pattern queries (Q(P), {H}) or retrieval queries (Q(R), {R}). Some queries are clearly linked. For example, a pattern query Q("cat") may return the hit set = {H34.57, H34.59a}, followed by the retrieval set {H34.59a}. A subsequent event with a query Q(H34.57) is probably related to the initial event (Q("cat"), {H34.57, H34.59a}), although this likelihood diminishes as time separates the events. The handling of patterns of inquiries could be challenging.

A logging scheme can record a number of items. Table I illustrates some common attributes that come to mind. Other aspects not yet identified may also be important. Certain attributes are invariant over a portion, or all, of a user's session. For example, during transmission over the network, a transaction may split into a number of network level IP packets, but these packets will all have the same source and destination addresses and ports. Moreover, the transaction will typically use a single network protocol such as ftp (for file transfer), http (for web access), smtp (for e-mail), etc. Furthermore, we assume that the query and subsequent retrieval sets contain categories taken from a finite set, thus making the topic space manageable. For example, in a medical application, a particular patient record might contain the following categories: the most recent physical measurements, heart test information, and previous (historical) medical information.

Figure 3 expands the earlier block diagrams, showing two different types of requests as a flow of data representations. Doctor 101 makes the first query, which asks for information about patient 237. In addition to basic health information, two additional items, heart and insurance records, are specified. Here Q(P) = Q(patient 237, heart, insurance), {H} = {R} = "the specified records". Administrator 1 is making a different kind of request. He is not interested in the record of a specific patient, but rather in summary data from the database regarding Medicare information; specific topics include the number of patients insured and the number with balances past due. Here Q(P) = Q(summary, Medicare patient insured, Medicare patient past due), {H} = 0 and {R} = "the specified summary data". In each of these requests, the information stored in the initial log for processing includes the query space information (Q(P), {H}, {R}), the topic space information (T1, … , Tn) along with all of the other fields from Table 1, although for brevity, not all are included in Fig. 3.

Topic Space.

The essential issue is a categorization of inquiries, responses and other logged information in a way useful to the host authority monitoring behaviors of users. In most of the envisioned situations, there is a large database of information. The goal in concern space is to characterize interests of individual query events in terms of management's concerns. This requires repository items (records, reports) to contain metadata in the form of a set of categories that are finite in number [3]. It can also apply to queries alone. A query Q("ebola") clearly matches the higher-level concern hemorrhagic fever. This occurs independently of any retrieval, R. The hemorrhagic fever classification correctly hints, however, that the source of categories is a challenge. One may have tens of thousands of web pages, speech samples, images, or medical records. Traditionally, a domain expert does labeling by examining each record separately. Given the large size of many modern databases, it is infeasible to assign each record manually. As new records (or queries) enter or existing ones change, some categories can be assigned based on the users' role when entering information and the database fields being modified. In applications with established keywords, keyword aggregation may be possible (e.g., family Filoveridae subsumes ebola). Alternatively, an expert can label a subset, say 10 or 20 percent, of the records, and an automated procedure attempts to label the rest. This action is a type of semi-supervised learning [1]. For now, we assume that an initial set of categories can be determined in some manner, with further research needed to refine the categories. Particular emphasis is upon evolving economical models that can work in everyday practice. The straw man assumes that Google-like page rank metrics are not available, largely because no assumption can be made about materials being hypertext (*i.e.*, having imbedded pointers to other materials). If the repository is hypertext, some elements seem more tractable; metadata matters less and the use of hypertext links can be monitored (could add this to Table 1).



Doctor 101	Patient 237	Heart Record	Insurance Carrier	•••

Administrator 1	Medicare Summary	Number Insured	Number Past Due	•••

Mapping (Table, Algorithm, etc.)
A priori information
Topic Space
Topics for Request 1

2	0	0	7	0	0	0	6	•••	0

0	0	0	3	0	11	4	0	•••	0

Concern Space
Bi–directional Mapping (Table, Algorithm, etc.)
Longitudinal Scores for User 1

371	0	11	976	5	3	2	642	•••	9

Longitudinal Scores for User 2

32	16	5	378	33	794	557	22	•••	7

	Status	Medical Specialization	Admin. Responsibilities	
User 1	Visiting	Cardiology	No	•••
User 2	Permanent	Internal Medicine	Yes	•••

Figure 3. Schematic of Mappings

Mapping of Query Space to Topic Space.

Discussion in the previous paragraphs has provided some background detail to the mapping process. As shown in Figure 2, a mapping must convert events in (Q(P), {H}, {R}) query space into a vector in topic space. The mapping takes into account any *a priori* information, which for a given record may include classification keywords, concordances, style and content scans of textual bodies [4]. The index pages for a book are another typical source. Queries can be handled with a suitable dictionary, although this is easier said than done. The goal is to determine likely categories (and implied concerns) for events in (Q(P), {H}, {R}). A single request may require data from multiple categories, and furthermore, that two different types of request may require data from the same category. A useful mapping is non-trivial. Determining appropriate mappings for a particular distributed medical database is one of the long-term challenges of this work. Some logs will become too large, requiring log summaries. The nature of such summaries is open.

Categorizing Users' Queries: Concern Space.

Given the set of log files detailing each query (Q(P), {H }) or (Q(R), {R }) in concern space, it is useful to consider transactions of a given user during some time epoch. Assuming, as discussed above, that each request contains a number of topics that can be grouped into a finite number of categories, one can then build a two-dimensional table listing the individual's request by date for each category. For our initial processing, we suggest reducing this data down to a single vector of dimension equal to the number of categories. As a first guess, about 100 categories would be a reasonable number; larger dimensionality can be handled, but it leads to sparse data and more difficult processing. Each element in this vector will then contain a nonnegative integer that represents the accumulated weight for this category from each event in (Q(P), {H}, {R}). This vector represents a users' "topic signature" over a specific time interval. Depending on the number of requests typically generated by a user, this signature can be developed on a daily, weekly or extended basis. Such a signature will vary over time—investigation of non-stationary signature behavior is an important aspect of such analysis.

Given topic signatures for users, one would like to determine if any signature is suspicious. Depending on the amount of metadata (*a priori* information), there are a couple of different processing options. The most straightforward approach is to aggregate users' signatures into some number of clusters. In [2], this approach is used for the related problem of aggregating network flows for the purpose of congestion control--each cluster is mapped directly to a queue, and a scheduling algorithm allocates bandwidth among these queues. Processing and interpretation is not so simple when categorizing users' behavior. There is no single one-to-one mapping between cluster and control action that is optimum in all cases. Initially, we propose using operator-controlled analysis of these clusters, with the aim of determining an interpretation for them. The clustering procedure can be done in a number of ways including through the use of the k-means algorithm, principal components analysis, or self-organizing maps [5]. In principal components analysis, the input data is linearly mapped to a feature space where the data is sorted by the amount of variance in each direction. Often, data reduction can be done in this feature space, by ignoring dimensions with low variation. Self-organizing maps are a non-linear generalization of principal components.

Since the clustering process does not require any *a priori* class information, it is a type of unsupervised statistical learning. In a typical medical records application, there will usually be additional meta-information that can be used to design supervised learning algorithms. Specifically, users may assume roles from a set of professional classes: doctor, nurse, orderly, administrator, insurance agent *etc.* Also, they can affiliate by department, such as accounting, radiology, dermatology, or cardiology. One assigns each request one or more labels from these categories. The result is a sequence (**x**[n], l[n]), n=1,…,N. **x**[n] is the vector describing transaction n, and l[n] is the corresponding set of labels. Given this training sequence, one can employ standard approaches such as decision trees [6], neural networks [7], as well as more recent approaches such as support vector machines [8]. Briefly, a support vector machine uses a hypothesis space of linear functions, and it operates in a high-dimension feature space obtained by a non-linear mapping of the data. The non-linear transformation and the inner product operation in the feature space can be combined through the use of a kernel; thus, support vector machines are in the class of kernel methods.

VI. Summary and Conclusions

All sensitive electronic record systems (ERS) risk insider-threats, those hidden, unanticipated ERS activities that constitute unacceptable use. These threats are—by definition—not detected through violations of individual access privileges. Several ERS communities—health care, finance and intelligence—share similar exposure risks from insider-threats. (Electronic publishing is another, in which attribution, permissions and in the end, considerations—payments, count heavily.) We have proposed a flexible general framework to promote discussion and progress on the ERS insider-threat. Our general framework introduces a perspective for short term behavior patterns and an intentional view for longer epochs of monitoring.

We have supplemented the general framework with a straw man that demonstrates challenges inherent in any expansion of the general framework. Our straw man employs concepts of query space, topic space and concern space. The straw man uses statistical learning methods to tease out elements of concern space over time. Query space and topic space of the straw man are, in principle, attainable, although much advanced development is needed to refine them and their various mappings. Concern space presents major research hurdles. Given an amorphous classification n-space, it is not yet clear how one correlates unlabelled classification volumes with known concerns and intentions, and conversely, what interpretations can be placed on arbitrary n-space volumes. Much effort is needed to understand how to apply various statistical learning techniques and to interpret their results in concern space.

References

[1]. D. J. Miller and J. Browning, "A mixture model and EM-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 25, No. 11, pp. 1468-1483, Nov. 2003.

[2] N. Chevrollier and R. E. Van Dyck, "Packet filtering for aggregate-based congestion control," *Proc. Conference on Information Sciences and Systems*, Princeton Univ., Mar. 17-19, 2004.

[3] J.C. French, *et al.* "Multiple viewpoints as an approach to digital library interfaces," *Proc., Workshop on Document Search Interface Design and Intelligent Access in Large-Scale Collections*, July 2002.

[4] P.W. Foltz. "Latent Intentional Analysis for text-based research. *Behavior Research Methods, Instruments and Computers* 28, 2(1996), 197-202. Available at www-psych.nmsu.edu/~pfoltz/reprints/BRMIC96.html

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*, Springer Verlag, 2001.

[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont Californis, 1984.

[7] S. Haykin, *Neural Networks A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, 2nd Ed., 1999.

[8] B. Schoelkopf and A. J. Smola, *Learning with Kernels Support Vector Machines*, *Regularization, Optimization, and Beyond*, The MIT Press, Cambridge, MA, 2002.

[9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, revised second edition, Morgan Kaufmann, San Francisco, 1997.