Processing Events in Probabilistic Risk Assessment

Robert C. Schrag, Edward J. Wright, Robert S. Kerr, Bryan S. Ware
Haystax Technology
McLean, VA USA

***Abstract*—Assessing entity (e.g., person) risk from entity-related events requires appropriate techniques to address the relevance of events (individually and/or in aggregate) relative to a prevailing temporal frame of reference—for continuous risk monitoring, a running time point representing "the present." We describe two classes of temporal relevance techniques we have used towards insider threat detection in probabilistic risk models based on Bayesian networks. One class of techniques is appropriate when a generic person Bayesian network is extended with a new random variable for each relevant event—practical when events of concern are infrequent and we expect their number per person to be small (as in public records monitoring). Another class is needed when (as in computer network event monitoring) we expect too many relevant events to create a new random variable for each event. We present a use case employing both classes of techniques and discuss their relative strengths and weaknesses. Finally, we describe the semantic technology framework supporting this work.**

***Index Terms*—temporal relevance; event relevance; anomaly detection; qualitative Bayesian network specification; probabilistic model; insider threat**

I. Introduction

Different parties have legitimate interests in understanding the risks that may be incurred when given persons are allowed to act in given roles. Employers are concerned about prospective employees, lenders about borrowers, landlords about tenants, and judges and parole boards about convicted criminals. To each role is accorded some privilege or stake—access to information/influence/reputation, finance, property, or liberty/public safety—that a bad actor could abuse or damage. While it's usually impossible to predict how a specific person *P* may behave in a given role *R*, an interested party *Q* may apply a probabilistic risk model *M* to available information about *P* to understand where *P*'s assessed risk may fall relative to other current or prospective players of *R*. *M* may:

- Have been derived from similar available data about other persons considered or selected for *R*
- Be based on legal or other policy doctrine
- Embody knowledge elicited from subject matter experts or published in a theory of human psychology.

Depending on the outcome of *M* applied to *P*, *Q* may choose to:

- Engage *P* in *R* (or not)
- Modify or disengage from its *R* relationship with *P*
- Invest more resources in assessing *P*'s risk (perhaps monitoring *P*'s actions)
- Counsel *P* in the positive management of factors related to *P*'s risk
- Modify *M* to accommodate an acceptable pattern of behavior not earlier addressed.

We have developed a series of related person risk models addressing the risk that *P* poses to *Q*. Each interprets the set of known events *E* involving *P* in assessing *P*'s risk. Each also must address the relevance of a given event *e* in *E* to computing *P*'s risk at a given time point *t* in *T*—the entire interval of relevant events (beginning, e.g., at *P*'s birth, majority, or engagement with *Q* and ending at the present or a most recently available event report date). Each model is probabilistic, calculating its core risk assessment using a Bayesian network (BN) [3]. Each model includes a generic person BN $\mathcal{B}$, which it may extend (based on *P*'s events) to create a person-specific BN $\mathcal{B}_P$.

A. Model M_C: Processing P's Life Events with Ingestion Rules

M_C addresses the risk that *P* may disclose *Q*'s private information without proper authorization, considering relevant event types (say, technical certification or conduct reprimand) that may collectively have a few instances per year. To address the effect of *P*'s events *E* on *P*'s risk, M_C runs a set of "ingestion" rules, each of which may extend $\mathcal{B}$ to reflect a given event *e* in *E*, ultimately resulting in $\mathcal{B}_P$. Because each triggered ingestion rule adds one or more random variables to $\mathcal{B}$, this approach tends to be practical when *E* is small relative to $\mathcal{B}$ (so that $\mathcal{B}_P$ does not grossly exceed expected computational requirements). M_C addresses the temporal relevance of a given event *e* by arranging that *e*'s influence on risk will build (when *e* is ongoing at *t*) or decay (when *e* is completed at *t*). This mechanism creates a conditional probability table associated with the temporal relevance of *e* to reflect the juxtaposition of *t* (the reference time for this risk calculation) with respect to the time point at which *e* occurs (if *e* is a point event) or the respective time points at which *e* begins and ends (if *e* is a durative event).

B. Model M_S: Processing P's Network Events with Summary Random Variables

M_S addresses the risk that *P* may pose an insider threat to *Q* via its access to *Q*'s information technology (IT) system—*Q*'s computers, computer networks, and related assets. The threat may be unauthorized information access, disclosure, theft, or destruction. M_S considers relevant event types (e.g., copying a file to a thumb drive or to an external website) that may occur many times per day. Given *Q*'s interest in assessing *P*'s risk on a continuous basis—over an employment interval—the "ingestion" approach that M_C uses to grow $\mathcal{B}_P$ with every event (instance) *e* in *E* is not feasible. Instead, for each such fine-grained event type τ, we include in M_S's version of $\mathcal{B}$ a random

variable (RV) summarizing the extent to which *P*'s actions are believed to warrant a suspicion of *P*'s exploiting *Q*'s IT assets towards insider threat. We calculate a likelihood for this summary RV so as to reflect:

- The relative novelty or familiarity of *P*'s events of type τ over:
  - *P*'s history
  - The synchronous history of other persons playing role *R* for *Q*
- Temporal decay.

M_S also considers relevant event types (e.g., copying a file to Wikileaks) that obviously manifest (vs. just warrant suspicion regarding) insider threat. For these latter event types, M_S uses the same ingestion approach as M_C.

C. *Model* M_G: *Combining* M_C *and* M_S

M_G combines M_C and M_S to address both the above aspects of insider threat—so that each model aspect can inform the other. E.g., if *P*'s non-network life events lead (the M_G version of) $\mathcal{B}_P$ to believe that *P* is likely **`Untrustworthy`**, this will increase (relative to a baseline, skeptical model state) $\mathcal{B}_P$'s belief that any potentially suspicious computer network actions actually do warrant suspicion. Also, staff members who warrant more insider threat suspicion on the computer network side receive higher overall risk scores, facilitating *Q*'s ability to react appropriately in general.

D. *Common Modeling Framework*

We have (following [5] and [6]) developed the generic person BNs $\mathcal{B}$ for the above models in our generic framework for large-scale probabilistic modeling that lets us specify:

- Boolean-valued RVs (generally, person attribute concepts—e.g., **`Trustworthy`**)
- Directed influences between concepts with discrete, qualitative strengths (obviating the usual BN requirement to specify—manually—for each RV a conditional probability table with one numeric entry for each element in the Cartesian product of its parents' domains—i.e., 2^n for *n* Boolean-valued domains)
- Generic modeling patterns for concept indication, mitigation, and relevance.

Our framework compiles such qualitative specifications into a representation executable by an off-the-shelf BN tool. (We use the Netica® API from Norsys.) Our $\mathcal{B}$ for M_C includes hundreds of RVs. Our specification of $\mathcal{B}$ for M_S is small enough to exhibit below (see Fig. 10, in the Appendix). $\mathcal{B}$ for M_G is again large—and the framework's support for layering of qualitative source specifications affords a much easier path to M_G than if we had built an M_S BN outside the framework.

Our ingestion rules that extend the generic person $\mathcal{B}$ into a person-specific $\mathcal{B}_P$ are described further in section VI.

E. *Sequel*

The sequel describes techniques we use to implement temporal relevance under the ingestion and summary approaches (introduced above with M_C and with M_S, respectively). We also exhibit results from the combined model, M_G, and discuss design trade-offs. Finally, we describe our supporting semantic technology framework.

II. Computing Temporal Relevance for Events Individually Ingested

Intuitively, the impact of a major life event on one's reputation is time-limited. While positive life events tend to build our confidence in a person—and negative ones erode it—the glow of accomplishment—like the stain of failure or breach—naturally fades over time. In our "whole-person" model M_C, we uniformly invoke exponential decay (or growth) with half life γ per an invoked ingestion rule Γ. The generic person Bayesian network (BN) $\mathcal{B}$ accounts for interactions among beliefs about random variables (RVs) representing different person attribute concepts like those in Fig. 1.

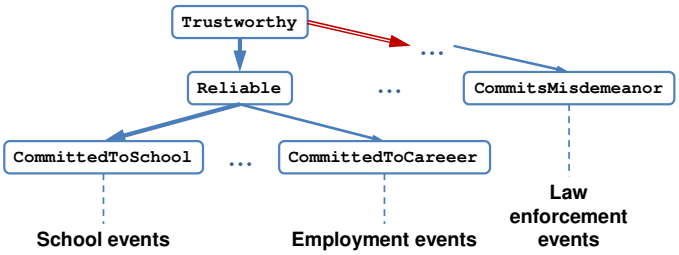

Fig. 1. Partial generic person attribute concept BN $\mathcal{B}$ (top), with related event categories (bottom).

In Fig. 1, BN influences point (causally) from indicated concept hypothesis to indicating concept. Stronger indications have thicker arrows, a single negative indication has a red, double-lined arrow. The full BN includes several hundred nodes (mostly elided).

M_C's ingestion rules apply *P*'s event evidence to $\mathcal{B}$ to develop a person-specific BN $\mathcal{B}_P$ including temporal relevance RVs (as explained next) appropriate for a given reference time point *t*. $\mathcal{B}_P$ then calculates the risk at *t*. By constructing $\mathcal{B}_P$ and calculating risk at successive time points, we develop a historical risk profile (i.e., a risk timeline) for *P*. See Fig. 2.

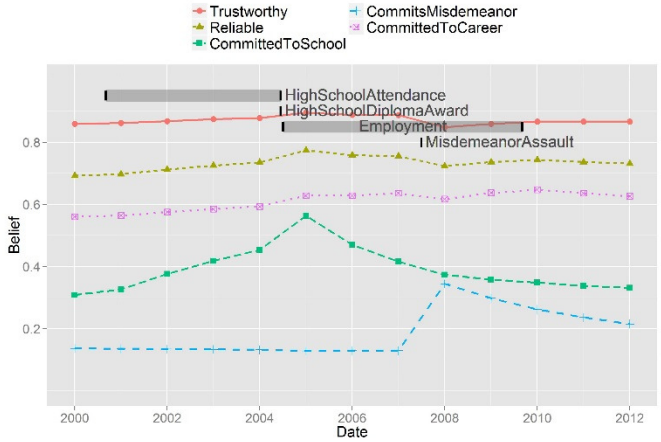

Fig. 2. Person risk timeline with life event overlay—per M_C.

In Fig. 2, *P*'s events are plotted in bars (top left to center). Belief over time is plotted for person attribute concept RVs per legend (top). **`Trustworthy`** is our top-level proxy for (the complement of) risk. Note how beliefs in **`CommittedToSchool`** and **`CommittedToCareer`** tend to build while the related (**`HighSchoolAttendance`** and **`Employment`**) events are ongoing. Influence interactions in $\mathcal{B}$

cause belief in **CommittedToCareer** to grow even while *P* is still in high school. (We tend to believe that someone who does well in school will also do well in a career.) Belief in **CommittedToSchool** increases when *P* graduates but then become less relevant per half lives specified in ingestion rules for school-related events. The 2007 **MisdemeanorAssault** charge decreases belief in all the other, positive concept RVs. See also Lisp macro calls expressing associated event data in Fig. 9.

When Γ ingests an event *e* (e.g., of type **EmploymentReprimand**), it instantiates a BN design pattern that:

- Creates an RV δ standing for *e* itself (an evidence RV)
- Creates an RV ρ standing for the temporal relevance of *e*
- Installs both ρ and the indicated person attribute RV π (standing for, e.g., **DisregardsEmploymentRules**) as BN parents of δ (see left panel in Fig. 3, below)
- Creates appropriate conditional probability tables (CPTs) for δ and ρ—denoted CPT(δ) and CPT(ρ).

CPT(ρ) encodes *e*'s nominal relevance at *t*, calculated per ingestion rule Γ's specified half life γ and the time α elapsed from *e*'s time point (designated by Γ as "beginning" or "ending," when *e* is durative) until the reference point *t*. For the case of relevance decay, we have $\theta = ½^{(\alpha / \gamma)}$. We specify θ as the probability P(ρ = *"true"*) and 1 – θ as P(ρ = *"false"*).

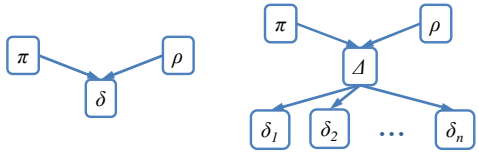

Fig. 3. Current ingestion scheme (left) and potential variant (right).

In Fig. 3 (left), BN influences are associated with an (ingested event) evidence random variable (RV) δ, an indicated person attribute concept π, and a temporal relevance RV ρ. In Fig. 3 (right), an evidence summarization scheme (using summary RV Δ) can insulate similarly-typed, closely temporally spaced events δ_i against departures from their nominally specified temporal relevance beliefs (say, θ_i) that otherwise would be induced among individual RVs (say, ρ_i)—had rather these been used.

CPT(δ) respects *e*'s strength (specified in Γ) as an indicator for π and arranges that the probabilities P(π = *"true"*) and P(π = *"false"*) observed in $\mathcal{B}$ for π do not depart (via the normal course of Bayesian influence propagation) from the nominal value θ installed in CPT(ρ). This is a local correction that is subject to further departures when other ingestion rule executions also modify $\mathcal{B}_P$. While we have observed this ingestion technique for temporal relevance to work well in practice, when two or more events in E are both semantically and temporally close to each other, we again see temporal relevance departures resulting from Bayesian influence propagation in $\mathcal{B}_P$. (The relevance RVs ρ_i tend to reinforce each other, amplifying their observed beliefs beyond their nominal θ_j. In some applications, this pattern may be appropriate; in others not.) We can ingest two nearly simultaneous (like-type) misdemeanor events without blatant departure from nominal θ. Ingesting five such events, we see θ decay only some 6% γ days after the events' occurrence (when we might naively have expected 50%).

We can—for quasi-simultaneous events—decouple the influence of temporal relevance from multiple indicating evidence events by invoking the alternative BN design pattern in in Fig. 3 (right panel), where Δ is a summary RV for individual event RVs δ_1, δ_2, … , δ_n. Accommodating evidence events δ_i occurring at materially different time points requires a more general approach to avoid the departures of temporal relevance beliefs from nominally specified values. The approach we describe in section III works well in this regard, but it does not afford the same expressive power as ingestion rules (which can consider arbitrary temporal relationships between events—as discussed in section V).

III. Temporal Relevance with Summarized Events

Computer network events that may inform *Q* about an insider threat by its engaged staff member *P* can occur so frequently that the ingestion technique described in section II is impractical. M_S avoids this issue by appealing to event summary RVs, as outlined in section I. See Fig. 10 in the Appendix, where the RV **CopyDecoyToExternal_Summary** (e.g.) summarizes the suspiciousness of actions in which *P* has committed the network action **CopyDecoyToExternal** (i.e., copy a seeded "decoy" file to an external location, such as a website).

Fig. 4 (full page) exhibits key metrics we compute for such a summary RV. Because we expect network event monitoring to be continuous—with practically unbounded beginning and no ending—we compute key metrics in temporal buckets of exponentially increasing size (top three charts)—so that we can always double temporal (if not event) capacity by adding one more bucket. Event occurrence buckets summarize (top/first) event type count since monitoring started, (second) signal with respect to *P*'s own history, (third) signal with respect to (a statistic computed over) the full/relevant monitored population. Each of 64 day's variation metrics (middle charts) are computed with respect to just the other buckets for that day, normalizing ratios of counts in related buckets to the range [0, 1] using a sigmoid function. Visual "floors" in the bottom three charts are set at 0.5 (the sigmoid function's point of symmetry)—highlighting direction of signal change. Variation with respect to own history compares events new in a bucket (those counted in the next-largest bucket) to those that are old—reverting to bucket-dependent defaults when no earlier events exist. To develop a suspicion likelihood (bottom chart), we first take a weighted average of each of the two anomaly metrics (with weights increasing, e.g., by bucket recency), then average these results. We then enter this suspicion likelihood (using the Netica® API) as a likelihood finding on a summary RV (such as **CopyDecoyToExternal_Summary**).

Under even weighting (invoked in Fig. 4 and Fig. 5), *e*'s relevance approximates $1/\alpha$. Compare this to our half life decay function (from section II) used under ingestion: $½^{(\alpha / \gamma)}$. Either class of techniques would in fact be compatible with either of these (or other) functions of relevance over time. We reviewed the overall half life approach and half lives appropriate for specific event types in M_C with experts in the subject matter of unauthorized information disclosure risk. Decay rates for computer network events in M_S have yet to be tuned in the context of real-world data.

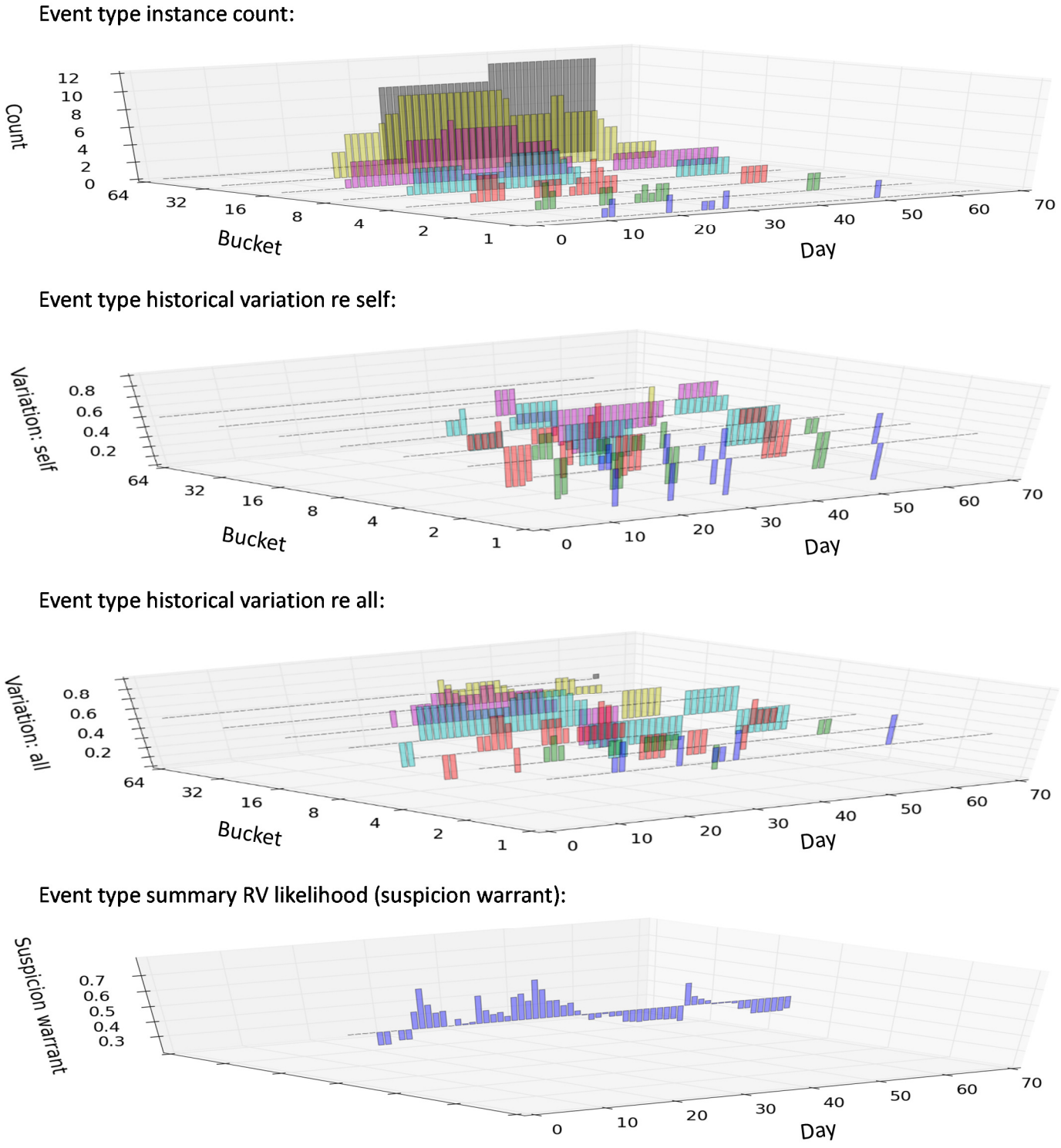

Fig. 4. Key metrics for a summary RV in an overall event type-related suspicion warrant.

Not yet exhibited in our processing in Fig. 4 and Fig. 5 is an approximating space optimization that would shift the "old" content (see Fig. 4) from bucket j into bucket $j + 1$ at each $(2^j)^{\text{th}}$ time step (for values j descending from the highest value for which $(t \bmod 2^j) = 0$—rather than computing the buckets' counts afresh at each time step, as shown. The latter approach would require retaining full event counts for all time steps—impractical for long-term, continuous operation.

A. Related Work

Senator et al. [4] describe a flexible insider threat detection framework providing statistical and machine learning components that may be applied to data across different time

scales. In contrast to our bucketed approach, their time scales must be explicitly specified by an application architect. They illustrate a component workflow motivated by a specific threat scenario. We have taken such scenarios to be compiled into random variables (for indicators and threats) in a Bayesian network (focused on overall risk assessment, rather than specifically on threat incident detection). Our framework can address a broad set of statistics in and over temporal buckets, supporting event processing over an arbitrary time scale. They describe results with a real-world dataset covering two months. We have developed our approach using a similar synthetic dataset [1] covering 18 months.

IV. Combined Model Use Case

We combine M_C with M_S—producing M_G—by appending the input models' influence graph specifications and defining M_S's **`Untrustworthy`** as the opposite of M_C's **`Trustworthy`**. As discussed in section II, this affords a path for P's non-network, life events to influence the risk measured for P's network events—thus enhancing the signal to noise ratio for persons who seem risky generally. See Fig. 5.

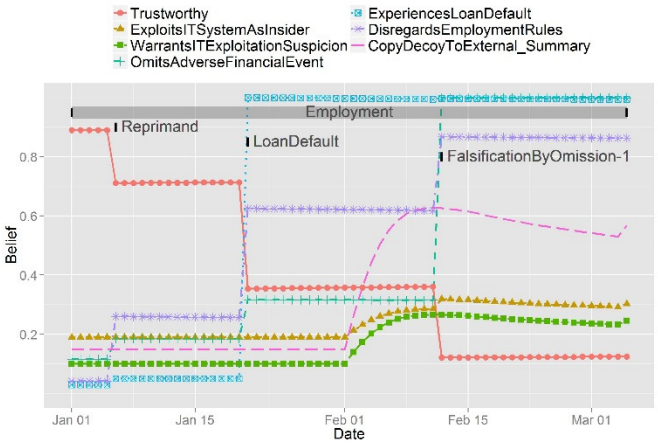

Fig. 5. Person risk timeline—per M_G.

In Fig. 5, life events (top, left to center) are per the M_C component. Belief in P's attribute concept RV **`Trustworthy`**—our top-level proxy for (the complement of) risk—falls lower with each successive derogatory life event (**`Reprimand`**, **`LoanDefault`**, **`FalsificationByOmission`**). This increases the relevance of **`WarrantsITExploitationSuspicion`** to **`ExploitsITSystemAsInsider`** in the M_S component. (This influence is not mitigated, because P is not engaged by Q in the role of IT administrator—see Fig. 10 in the Appendix.) Belief in **`WarrantsITExploitationSuspicion`** (third belief line from bottom on plot's right) takes a jump of about 10% when M_S's summarized events mount (level until day 33, then increasing by one counted event per day) and M_C's ingested events have occurred. By comparison (not shown), belief in **`WarrantsITExploitationSuspicion`** jumps by only about 3% when M_S stands alone, uninformed by an M_C component. Life events decay per half lives not conspicuous at the depicted time scale.

M_G is a proof of concept. Additional cross-model linkage and tuning of relevant modeling parameters may further increase the signal to noise ratio for network user risk detection when non-network, life events are consider in the same unified model.

V. Discussion: Event Processing Design Trade-Offs

The different domains we have addressed in M_C and M_S have presented event processing requirements largely amenable to—in fact, engendering—the two classes of techniques described here: per-event ingestion (section II) and event summarization (section III), respectively. As noted in sections II and III, per-event ingestion is liable to (possibly unintended) amplifications of temporal relevance, when event instances are both semantically and temporally close. This is, however, just the situation for which we have designed event summarization. While Fig. 3 (right) suggests a hybrid approach for M_C to aggregate events that are temporally close, this really begs the question: *What should be the effect of similar indicating events on the belief calculated for an indicated hypothesis random variable (RV) in a Bayesian network (BN) $\mathcal{B}$?*

Event summarization, in M_S, adopts the extreme position that all events of a given type τ should be summarized in a single random variable (RV). The uniformity of this approach may be appealing, but it bears a simplicity driven by the necessity of addressing a practically unlimited stream of fine-grained events—many of which are relatively weak indicators of insider threat. Event ingestion, in M_C, hazards (so far, in our application domain, rare) potential amplifications in temporal relevance but affords the power (via Allegro Prolog®-based ingestion rules and auxiliary predicates—see section VI.A) to express nuanced temporal configurations of events of different types that are not obviously amenable to bucketed historic summaries. We might note temporal overlaps to extract certain compound events (say, of type **`FailsDrugTestDuringEmployment`**), but we could not refer to earlier relevant events, as in Fig. 6.

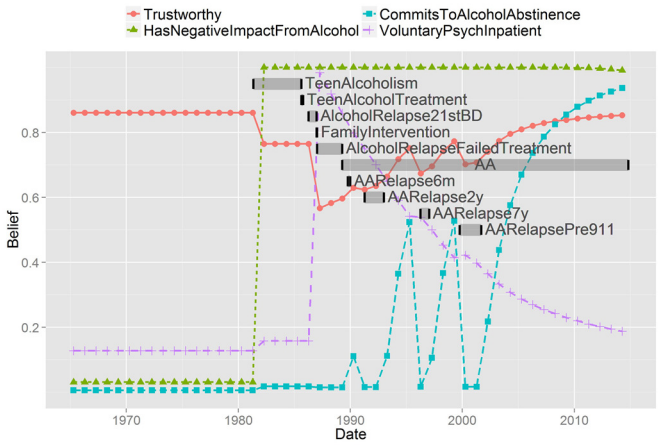

Fig. 6. Ingestion rule processing of non-overlapping events in M_C.

In Fig. 6, P, with a history of alcoholism, has an ongoing commitment to alcohol abstinence but also several intervening alcohol problem events (relapses). While M_C considers an alcohol abstinence commitment to mitigate an earlier alcohol problem, this commitment is voided (and the building of P's credibility begins again) when P relapses. In M_C, the temporal specificity of a mitigating event (or generality of a mitigating person attribute concept) is important in determining whether to capture this effect with an ingestion rule, with a **`:MitigatedBy`**

influence specification (see Fig. 10 in the Appendix), or with a combination of such mechanisms. M_C's related ingestion rule invokes our Allegro Prolog® predicate **mostRecentLaterStartingReportedEvent** to rebase the temporal relevance computation for the person attribute concept **CommitsToAlcoholAbstinence** at a most recent relapse event's ending point.

Intimately related to realizing an appropriate overall semantics for a person risk rating model M (but outside the scope of the present discussion about processing person events) is the design of person attribute concept RVs and concept-to-concept influences in a generic person BN $\mathcal{B}$. Even before we decide how to associate encountered evidence E with $\mathcal{B}$, we must be happy with $\mathcal{B}$'s inferences under arbitrary (likelihood or domain value) findings for $\mathcal{B}$'s RVs. This requires thinking (and testing) at least as hard about the semantic relationships among person attribute concept definitions and connecting influences as we do about those among event types and processing styles. (See also section VI.C.)

VI. Supporting Semantic Technology

Our ingestion rules are written in Allegro Prolog®. They read events expressed using an OWL ontology from an AllegroGraph® triple store and create a person-specific BN $\mathcal{B}_P$ using the Allegro Common Lisp® API to the Netica® API. Allegro Prolog®, AllegroGraph®, Allegro Common Lisp® are products of Franz, Inc. Netica® is a product of Norsys, Inc. The Allegro Common Lisp® API to the Netica® API is open-source. We see unique benefits in this software stack.

A. Ingestion Rule Design

AllegroGraph® is an RDF triple store management system that happens to be written in Allegro Common Lisp®. While Franz supports AllegroGraph® clients for a number of different languages, the direct (vs. remote) Lisp client benefits us in that it shares memory with AllegroGraph® itself. Allegro Prolog®, written in and included in Allegro Common Lisp®, is a logic programming facility that the Lisp direct client extends with Lisp macros and Prolog predicates affording access (alternatively to SPARQL) to AllegroGraph® triple stores. Because Allegro Prolog® supports calls to Lisp functions from within logic programming rules, our ingestion rules can invoke the Allegro Common Lisp® API to the Netica® API to augment an existing generic person Bayesian network (BN) model $\mathcal{B}$ to add random variables (RVs) corresponding to a person P's events E, resulting in a person-specific BN $\mathcal{B}_P$. See Fig. 7.

```
(defIngestionRule RestrainingOrder
    (+process-reportedEvent ?person ?*asOfDate)
  (reportedEvent ?person
                 ?*asOfDate
                 ?event
                 !agent:ProtectiveRestrainingOrder
                 ?*startDate
                 ?*endDate
                 ?*ongoing?
                 ?*reportDate)
  (lisp (create-EventConceptIndication
         ?person
         :IndicatedConcept CommitsDomesticViolence
         :+IndicatingEvent ?event
         :Terminus :end
         :DeltaDays (- ?*asOfDate ?*endDate)
         :HalfLife (* 6 365)
         :Strength :strong
         :Polarity :positive)))
```

Fig. 7. M_C ingestion rule.

In Fig. 7, **RestrainingOrder** names an ingestion rule pertaining to events of type **https://haystax.com/agent#ProtectiveRestrainingOrder** (whose RDF namespace part is signaled in AllegroGraph® by the prefix **!agent:**). **defIngestionRule** is a macro wrapping Allegro Prolog® **<-**, registering the ingestion rule and performing static analysis to ensure well-formedness. **+process-reportedEvent** is the predicate—of which all ingestion rules are members—used to launch ingestion rules for a given person and reference time. Logic programming variables are prefixed by **?**, Common Lisp keywords by **:**. We use the prefix **?*** as a convention noting that a binding should have a native Lisp value, rather than an RDF part (i.e., a resource or a literal). **?person** and **?asOfDate** will be bound when the rule is called. The call to **reportedEvent** succeeds when **?event** can be bound to an instance of **!agent:ProtectiveRestrainingOrder**, such that **?person** is the **!agent:riskRatingSubject** of **?event**, and **?event**'s temporal characteristics and provenance are appropriate (binding values for other logic variables). When **reportedEvent** succeeds, the call to the Lisp function **create-EventConceptIndication** is executed, augmenting $\mathcal{B}_P$ as explained in section II and illustrated in Fig. 3 (left): **CommitsDomesticViolence** takes the role of ρ, **?event** induces the new random variables δ and π, the value of the **:DeltaDays** keyword argument takes the role of α, that of the **:HalfLife** argument the role of γ. The ingestion rule itself serves as Γ. Upon completed ingestion processing, the realized $\mathcal{B}_P$ can be compiled and queried for beliefs in person attribute concepts of interest.

Allegro Prolog® includes predicate-level functors supporting logical operations (e.g., **and**, **or**), backtracking control (varieties of **if**, cut), and Lisp calls evaluated at predicate level for their truth values (i.e., not just execution for side effect as in Fig. 7). Under AllegroGraph®'s direct Lisp client, user-defined Allegro Prolog® rules (so ingestion rules and their supporting predicates) may include any RDF resources (i.e., URIs) or literals.

B. Event Ontology Design

With its signature treatment of programs as data (both expressed as lists), Lisp has long been a favorite language for

creating embedded knowledge representation languages and supporting utilities. We exploit this facility in designing our models' ontologies for person-related events—using Lisp macros to express class, property, and individual (instance) definitions. See Fig. 8.

```
(defOntologyClass Person (Thing)
  (hasGender Gender :Functional))

(defOntologyClass Gender (Thing)
  (:enumeration Male Female OtherGender))

(defOntologyType Date !xsd:date)

(defOntologyClass Event (Thing)
  (riskRatingSubject Person :Functional)
  (startDate Date (:cardinality 1))
  (endDate Date :Functional)
  (sourceReport Report :Functional))

(defOntologyClass PointEvent (Event)
  (hasConsequentEvent Event))

(defOntologyClass DurativeEvent (Event)
  (hasSubEvent Event))

(defOntologyClass ProtectiveRestrainingOrder
    (PointEvent))
```

Fig. 8. Lisp macro calls defining elements of our event ontology for M_C.

Macro calls in Fig. 8 add triples to a specified graph in an active store. Store-resident triples may be serialized to a standard OWL file in (e.g.) RDF/XML format, then viewed in an available ontology browser (e.g., Protégé). RDF namespace designations (e.g., **!xsd:**) are required only where these deviate from a specified default. For a specified class (e.g., **Person**), an object or datatype property (e.g., **hasGender** or **startDate**) is created per the type (e.g., **Gender** or **Date**) specified. OWL closed enumeration classes (e.g., **Gender**) are supported, as are OWL property types (e.g., **Functional**) and restrictions (e.g., **cardinality**). Validation machinery ensures a specified ontology's global consistency with respect to effective cardinalities allowed.

Per Fig. 8, we now have a single-actor event ontology. While M_S defines persons' roles (e.g., system administrator) with respect to organizations, we have not yet broached persons' roles in events beyond the person-indexing property **riskRatingSubject**.

Our framework also has a Lisp macro useful for defining hand-crafted datasets. See Fig. 9.

```
(defOntologyInstance !data:P (Person))

(defOntologyInstance !data:PHighSchoolAttendance
    (SchoolAttendance)
  (riskRatingSubject !data:P)
  (schoolCredentialAward !data:PDiplomaAward)
  (startDate "2000-09-04")
  (endDate "2004-06-15"))

(defOntologyInstance !data:PDiplomaAward
    (SchoolCredentialAward)
  (riskRatingSubject !data:P)
  (startDate "2004-06-15")
  (schoolCredentialAwarded HighSchoolDiploma))

(defOntologyInstance !data:PEmployment
    (Employment)
  (riskRatingSubject !data:P)
  (startDate "2004-07-05")
  (endDate "2009-09-05"))

(defOntologyInstance !data:PMisdemeanorAssault
    (PoliceOffense)
  (riskRatingSubject !data:P)
  (offenseChargeSchedule Misdemeanor)
  (startDate "2007-06-30"))
```

Fig. 9. Lisp macro calls used to create the (minimal) dataset for the person profiled in Fig. 2.

The framework validates any loaded dataset with respect to declared subject and object classes, literal data types, and property types (e.g., **Functional**) and restrictions (e.g., **cardinality**).

C. Probabilistic Ontology Design

We do not now break down person attribute concepts (e.g., **Trustworthy**) beyond their status as such. Conceptually, they are properties of **Person** that—via their corresponding random variable (RVs) in $\mathscr{B}$—constitute a (flat) probabilistic ontology [1]. Relationships among these RVs are of the kind specified in Fig. 10. Most person attribute concept definitions in M_C include citations to and/or excerpts from guiding policy documents regarding information disclosure risk that also specify related indicating, mitigating, and relevance-inducing concepts.

VII. Conclusion

We have described two classes of techniques for processing events in probabilistic person risk models, examining the advantages and disadvantages of techniques in each class. Our proof-of-concept (section IV) combination of techniques from both classes demonstrates how inferences informed by either class of event processing can inform the other effectively. The selection of event processing techniques is one key element of overall risk model design, along with event ontology design and influence network design. In support of this work, we have developed and exploited appropriate semantic technology, with an eye towards flexible reuse.

Disclaimer

The views expressed are those of the authors and do not reflect the official policy or position of any legally recognized body or its representative parts or members.

References

[1] P. Costa and K. Laskey. "PR-OWL: A framework for probabilistic ontologies." Frontiers in Artificial Intelligence and Applications 150 (2006): 237.

[2] J. Glasser and B. Lindauer. "Bridging the gap: A pragmatic approach to generating insider threat data." IEEE Security and Privacy Workshops, 2013.

[3] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Series in Representation and Reasoning, 1988.

[4] T. Senator et al., "Detecting insider threats in a real corporate database of computer usage activity," Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013.

[5] C. Twardy, E. Wright, S. Canon, and M. Takikawa, "Credibility models," in Proceedings of the 5th Bayesian Modeling Applications Workshop. Vancouver, 2007.

[6] E. Wright and K. Laskey, "Credibility models for multi-source fusion," Proceedings of the 9th International Conference on Information Fusion, July 2006.

Appendix: Qualitative Specification for the Generic Person Bayesian Network in M_S

The generic person Bayesian network (BN) $\mathscr{B}$ for M_S is as specified in Fig. 10. "**_Summary**" random variables (RVs) correspond to an event schema derived from US CERT synthetic dataset "r6.2" available at http://cert.org/insider-threat/tools/index.cfm [1].

In Fig. 10, the top-level RV **ExploitsITSystemAsInsider** is a disjunctive summary of two next-level RVs: **CommitsITExploitation** (P has committed an unambiguously exploitative event—presumed rare) and **WarrantsITExploitationSuspicion** (covering network events that may or may not be exploitative). **CommitsITExploitation** is absolutely indicated ("implied") by three intermediate RVs that are in turn indicated (with varying strengths) by computer network event summary RVs. M_S enters likelihood findings for the latter into $\mathscr{B}_P$ as described in section III. **WarrantsITExploitationSuspicion** is considered relevant (absolutely) to **CommitsITExploitation** if P is **Untrustworthy**, mitigated (strongly) if P **HasRole-ITAdmin** (in Q).

```
(defparameter *Influences*
  '((ExploitsITSystemAsInsider
     (:ImpliedByDisjunction
      (CommitsITExploitation
       (:ImpliedBy (DestroysInformationUnauthorized)
                   (AccessesInformationUnauthorized) ; Ingested: HandlesKeylogger_Event
                   (DisclosesInformationUnauthorized) ; Ingested: CopyFileToWikileaks_Event
                   (StealsInformation))) ; Ingested: CopyFileToCompetitor_Event
      (WarrantsITExploitationSuspicion
       (:ImpliedBy (WarrantsInformationDestructionSuspicion
                    (:IndicatedBy (:Strongly (DeleteFileOnOthersPC_Summary))
                                  (:Moderately (DeleteFileOnLabsPC_Summary))))
                   (WarrantsUnauthorizedInformationAccessSuspicion
                    (:IndicatedBy (:Moderately (AfterHoursLogin_Summary))
                                  (:Weakly (OpenFileOnOthersPC_Summary))))
                   (WarrantsUnauthorizedInformationDisclosureSuspicion
                    (:IndicatedBy (:Strongly (CopyOthersFileToThumb_Summary)
                                             (CopyDecoyToExternal_Summary))
                                  (:Moderately (OpenDecoyFile_Summary)
                                               (AcquireDecoyFile_Summary)
                                               (CopyFileToExternal_Summary))
                                  (:Weakly (CopyFromThumbToOwnPC_Summary)
                                           (CopyOwnFileToThumb_Summary)
                                           (CopyOthersFileToExternal_Summary)))))
       (:RelevantIf (:Locally (:Absolutely (Untrustworthy))))
       (:MitigatedBy (:Locally (:Strongly (HasRole-ITAdmin))))))))))
```

Fig. 10. Qualitative specification for probabilistic influences in M_S (with semi-colons prefixing comments in red.)