Unified Architecture for Integrating Intelligence Data

Suzanne Yoakum-Stover, Ph.D.
Potomac Institute for Policy Studies, Senior Research Fellow
US Army CERDEC I2WD, Information Exploitation Futures Lab, Lead Scientist

Tatiana Malyuta, Ph.D.
New York City College of Technology, Associate Professor
US Army CERDEC I2WD Information Exploitation Futures Lab, Knowledge Manager

Problem Context and Statement

- Business of Intelligence
  - To develop and communicate understanding

- Intel Business Processes
  - Move Intel artifacts with respect to the cognitive hierarchy
    - Into: Data collection
    - Up: Semantic enhancement & fusion ➔ Information & knowledge
    - Out: Communication and collaboration ➔ understanding

- Data Integration Problem
  - Integrate all Intel into a coherent repository of knowledge
  - In an Ultra-Large-Scale systems environment
    - Decentralized
    - Inherently conflicting, diverse, and unknowable requirements
    - Heterogeneous, changing, and inconsistent elements
    - Normal failures, continuous operation, evolution, and deployment
    - Immense scale along many dimensions
  - Without attempting to control
    - Data sources, types, data-models
    - Processing, usage, application

Current Practice Fails

Merging or harmonizing data models, either physically or virtually, fails to accommodate the demands of the fluid and rapidly growing intelligence enterprise

- Physical integration of disparate models into a single canonical data-model is untenable in the face of scale and complexity and cannot adapt as the system evolves.
- Virtual integration lacks authority over data sources and fails to support inter-source collaboration without introducing yet another database.

What begins as a neat solution for a handful of systems quickly becomes intractable with scale. This phenomenon is but one early symptom of our evolution toward Ultra-Large Scale (ULS) systems and as such, invites a completely different approach - one that remains viable in a freely evolving, interdependent collective of systems, people, policies, cultures, and economics, very little of which will ever be under our control.

New Approach

- Our approach to integrating Intelligence data in a ULS systems environment is data-centric (as opposed to data-model – centric) and proceeds in two stages
  - The first addresses the unified storage of the entire spectrum of intelligence artifacts regardless of modality or representation.
  - The second stage builds upon the foundation provided by the first to address the unified storage of structured data to enable semantic data integration.
- The result is a layered data architecture that can accommodate any kind of data without placing restrictions on vocabulary, structure, semantics, or constraints, in a way that addresses the needs of the Intelligence Community today while providing a seamless transition path toward a future of ULS systems imbued with semantic technologies.
**Design Tenets**

- Layer 1 of our data integration architecture supports an aspect of collection and rudimentary exploitation. Layer 2 supports the processing by which data is enhanced with semantics to produce information, and the processing by which information is enhanced with richer associations to produce knowledge.

- We embrace the diversity of domain-specific data-models employed throughout the Intelligence Community by taking a data-model agnostic approach wherein the integration model makes the least possible commitment to any particular data-model.

- The character and meaning of the source data-model, when existent, is preserved and made accessible by the data store.

**Layer 1: Indigenous Artifacts**

- In Layer 1 we seek to integrate the entire spectrum of indigenous artifacts by collecting them in one (possibly distributed) database using standard means for physical and or virtual data integration.

- Crucial principles
  - Avoid making any data or data-model transformations in the process of data ingestion
  - Make the least possible commitment to a data-model in the target storage schema

Consequently, the Layer 1 database schema is quite simple and flat, exposing a minimal set of essential meta-data fields whose main purpose is to support back-tracking to the original artifact and or source.
Layer 1: Universal Indigenous Store

- Provides a manageable yet powerful and standard interface to the source data
- Gives us the option to either “lazily” load and cache data as “virtual artifacts” for performance sake, or persist and control data as “tangible artifacts” for the long term
- Provides “one stop shopping” access to the indigenous data for analysts
- Establishes a foundation upon which deep data integration can be more effectively pursued

Layer 2: Universal Store for Structured Data

The challenge—a universal storage model for structured data
- To accommodate structured data in a way that exposes that structure for use, without imposing the structure on the data store itself
- Determine a method for storing and managing any kind of structured data, reflecting any data-model, so that it can be shared, efficiently exploited, and extended in unforeseen ways without requiring model-specific storage implementations
The Problem with Structured Data

(a) Unstructured Data

(b) Data-model

(c) Structured Data

The data-model is imposed on the database and the data is frozen into it.

(d) Typical database structure

Layer 2: Data Model Abstraction

A domain-neutral storage model for structured data

- Decoupling that which varies, namely vocabularies and, more generally the data-models, from that which remains constant, namely the source artifact, and ideally the storage structure
- Considering structure, vocabulary, semantics, and constraints from a higher level of abstraction from which we then distill a minimal set of elements sufficient to capture any data-model
Layer 2: Elements

- **Mention**: A chunk of data, either physically located within a tangible artifact, or contained within an analyst’s mind
- **Concept**: An abstract idea, defined explicitly or implicitly by a source data-model
- **Predicate**: An abstract idea used to express a relationship between “things”
- **Term**: A disambiguated *mention* abstracted from the source artifact or asserting analyst
- **Statement**: Encodes a binary relationship between a subject and an object mediated by a *predicate*

Layer 2: Data
Layer 2: Data Model

Layer 2: Semantics
Layer 2: Semantic Associations

Data Description Framework (DDF)

The Layer 2 elementary constructs (concept, predicate, mention, term, and statement) provide the fixed-points of a data reference model that will ultimately serve as a practical data integration platform. We call this reference model the Data Description Framework (DDF).

Despite its simplicity, the DDF is a rich model that can be viewed from at least two different perspectives as a synergistic combination of two higher order models lying along different dimensions of abstraction

- **Extrospective**
  - Concept and predicate look outward toward domain knowledge.
  - Mention looks outward toward the data.

- **Introspective**
  - Term and statement form a semantic model and abstract data-model internals to expose structure in a uniform way.
DDF: Vertical and Horizontal Integration

Together the introspective and extrospective models enable both horizontal and vertical data integration:

- The extrospective abstraction bridges data and domain knowledge (vertical integration).
- The introspective abstraction bridges data structured by various disparate processes (horizontal integration) and binds the two outward looking faces of the extrospective model to provide a comprehensive data integration model.

DDF: Simply Put

- Useful integration results just from putting data in the DDF
- Mostly automatic process
  - Data of interest selected from external data stores
  - Automatic load into DDF
  - No data-model harmonization
  - No information is lost
- Queries on Terms
  - What is 7182605184?
  - What sources mention 7182605184?
  - What of the Locations mentioned in DB-A are also mentioned in as Locs in DB-B?
DDF: Stating the Obvious

- Relations in source data automatically become statements
  - Only small sample illustrated
  - No data-model harmonization required
  - No information is lost
- Queries on Statements
  - Capability equivalent to that of the source system
  - Examples
    - What terms, concepts, or mentions are associated via the predicate hasName?
    - What phoneAccs hasName Tanya?

DDF: Data Integration

- Nontrivial data integration by
  - Adding predicates
  - Creating statements that span across sources
- Enables
  - Correlation across data sources
  - Knowledge enhancement
  - More sophisticated queries
    - What are the PhoneAccs of those who work with Tanya?
    - What other labels does New York have?
Above and Beyond (Layer 3)

Connecting the Dots
- Halos represent distinct source systems.
- Associations
  1. Black: Automatic from ingestion into Layer 2
  2. Red: Added in Layer 3 to harmonize data-model elements
  3. Blue: Indicate data match, due to 2
  4. Green: Automatic result of 1-3
- Data in B used to generate new association between data in A and C (Green).

Conclusion

- We have presented the first two layers of a multi-layer data integration architecture that enables deep semantic data integration in a ULS systems environment.
- The underlying model, the DDF, supports both horizontal and vertical data integration (i.e. across disparate data-models and from data to knowledge) by embracing the diversity of data / knowledge models and processes by which data is structured.
- More importantly, the model admits a practical implementation ( "hard running code") that accommodates artifacts of any modality (e.g. text, audio, images, video, signals) in a single unified data store that enables true multi-intelligence data fusion and the continuous enrichment of data into knowledge.