Multi-INT Complex Event Processing using Approximate, Incremental Graph Pattern Search

Dr. Jim Law and Dr. Scott McGirr
SPAWAR Systems Center Pacific
San Diego, CA, USA

POC: Jim Law
619-553-2449
jim.law@navy.mil
Technical Approach

Problem: Dynamic Data
Constantly changing data from many sources leads to fragmented, inefficient search methods.

Approach: Characterize and Investigate
1. Observe actual composition and structure, and define nature of changes.
2. Investigate areas of theory that can accommodate observations.
3. Investigate initial feasibility.
Data Environment

Multi-Source Intelligence (Multi-INT)
- Images
- Video
- Comms
- Geo
- Human
- RF
- ...

Processing
- E/R Extraction
- Sentiment
- Tracking

Common Access Storage
- Federated Storage in RDF* Format

RDF: Resource Description Format (W3C Std.)
http://www.w3.org/TR/rdf-primer/
The Event Continuum

Three distinct regimes in the literature. Complex Event Processing in AIW event streams rarely has the characteristics of Static Networks.
Characterization:
• Variable temporal persistence.
• Independent events.
• Complex dependences.
• Evolution.
• Sequential events.
• Concurrent events.

Investigate:
• Dynamic graph theory.

Initial feasibility:
• Algorithm prototype.
Prototype Implementation

A Framework and Algorithms to infer complex events from high-volume streams of events in near-real time.

Technical Approach

The approach involves the following steps:

1) **Event Streams**: Investigate structure and phenomenology of EIW events and event streams. Augment streams by adding context, time stamps, pedigree and graph structure.
2) **Data Framework**: Tag and encode the data using lexicons, schemas, and ontologies. Investigate dynamic graph theory towards scalable graph update and search.
3) **CEP Algorithms**: Develop and implement new dynamic graph algorithms for approximate graph pattern search. Evaluate algorithms to identify relevant patterns, activities, and events.
4) **MOP Evaluation**: establish means to measure and improve performance.
Graph Pattern Match

Input: Pattern $P=(V_p, E_p)$, Data Graph $G=(V, E)$, and Ontology $O$

Initial step: compute all-pairs-shortest path matrix $M$.

A. Find descendent nodes, $D_{V_p}=\text{desc}(V_p)$, in ontology of each pattern graph node.

B. Compute potential matching set in $G$ for each node in $D_{V_p}$.

C. Traverse paths in the matching set, examining path length and edge type. Remove nodes that are not connected, do not meet path constraints, or do not have correct edge type.

Distance matrix is updated as graph changes using algorithms from Ramalingam, 1996.
IIMEF Exercise: Key Leader Engagement (KLE)

US FOB
Lat/Long 34.6720 -77.2402
MGRS 18STD9474638954

NAI 1 (JAFARNI VILLAGE)
Dynamic Data Graph is built incrementally, as information becomes available. Threat patterns can be detected at any time.

IPB: Intelligence Preparation for the Battlefield.
Watch for a Person with a previous involvement in IED Activity, and access to Fertilizer, a Vehicle, and a Storage Facility.

Given some hierarchy of objects and activities:

Watch the data as the graph changes and warn of any matching pattern.
Initial Performance Comparisons

Total execution time for 10 executions of 5 random pattern searches in synthetic data sets.

Graph Pattern algorithm prototype:
- 680 line implementation in Python
- Uses networkx graph library.
- Uses rdflib RDF library.

SPARQL query prototype:
- 200 line implementation in Python
- Uses rdflib RDF library.
- Uses rdflib SPARQL library.

Execution environment: Dell Precision T1500, Core i7 processor, 8GB RAM, Windows 7 OS, 1TB HD.
Results

- Completed initial studies of EIW data streams and content.
- Completed initial prototype of graph pattern search algorithms.
- Completed initial performance comparisons with std. search algorithms (*Naïve implementation was no worse than current standard algs, with improved capabilities*).
- Developed graph encoding framework, including activity and event hierarchy.
- Developed preliminary methods of performance assessment for activity discovery.
- Implemented prototype incremental graph.
Issues

Data issues:
• Access to suitable data with documentation.
• Relatively small data set sizes.
• Time constraints for processing and encoding data.
Future Work

• Mature graph encoding work for wider application and greater robustness.
• Automate conditioning of data to facilitate larger-scale testing.
• Participate in user experiments, when/where possible.
• Investigate additional application areas with larger data sets.