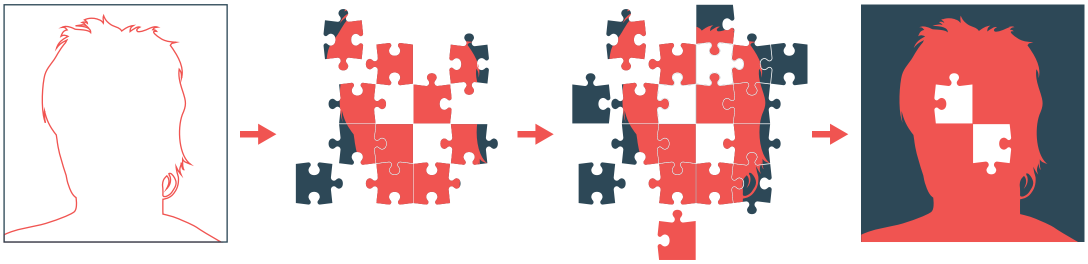


The Implementation of Complex Event Processing as a Behavioral Analytic

COGILITY

Introduction

> **Event processing** is a method of tracking and analyzing (processing) streams of information (data) about things that happen (events),[1] and deriving a conclusion from them. **Complex event processing**, or **CEP**, consists of a set of concepts and techniques developed in the early 1990s for processing real-time events and extracting information from event streams as they arrive. The goal of complex event processing is to identify meaningful events (such as opportunities or threats)[2] in real-time situations and respond to them as quickly as possible.[3]

Cogynt, by Cogility, is a Continuous Intelligence (CI) Platform that helps analysts make timely and informed decisions. Cogynt employs an advanced form of Complex Event Processing (CEP) that applies hierarchical complex event processing (HCEP). HCEP serves as the analytic core for the Cogynt CI platform that distills big data into actionable intelligence. The actionable intelligence is delivered in context, with risk—or opportunity assessment—and full provenance to assist decision makers. The Cogynt CI Platform is currently being applied to Insider Threat Programs for the Department of Defense (DoD) as a complete analytic solution that produces insider threat assessment insights from observed behavioral and technical indicators that are deemed to be potentially harmful to an organization. In many cases, a single indicator isn't sufficient to fully determine risk, rather, a combined set of indicators that represents a pattern of concerning behaviors, such as being passed over for promotion, disgruntlement, and law enforcement infraction, serves as an early warning of potential insider threat risks. The ability for the organization to identify potential insider threats early can benefit both the organization and its staff by promoting a safe and productive workplace. For an individual who is experiencing personal difficulties, the organization can act to help reduce risk by offering individual help—such as employee assistance programs. Proactive monitoring and analysis will also improve organizational security by mitigating future insider threat incidents.

In addition to its application in support of DoD insider threat programs, Cogynt is being applied in several commercial cyber use cases involving 3rd party risk assessments at exceptionally large scales (as many as thirty-one million entities[4]) on a continuous basis, reducing cyber risk for companies and their business partners. In addition, Cogynt is in the early phase of being applied to cyber insurance underwriting where there is huge promise of reducing loss-ratios for insurance companies that have adopted the technology.

In each of these use cases, there is a critical need to detect specific events in dynamic context while processing massive volumes of data and maintaining complete stateful knowledge of behavioral patterns for many thousands or millions of entities over prolonged periods of time. HCEP applies the concept of abstraction hierarchies to abstract low-level event data into higher-level sets of events reflecting actionable intelligence, which are more easily consumed by human analysts.

This paper explains the motivation for developing HCEP on a real-world use case, and then describes how HCEP works, why it is unique, and how it compliments other analytics such as artificial intelligence (AI)[5] / machine learning (ML).[6]

[1] https://en.wikipedia.org/wiki/Complex_event_processing#cite_note-LuckhamD-1
[2] https://en.wikipedia.org/wiki/Complex_event_processing#cite_note-Bates-2
[3] https://en.wikipedia.org/wiki/Complex_event_processing
[4] An entity is a person, place or thing. In the context of Cogynt, it typically references objects that have behavior.
[5] For further reading on AI please refer to https://en.wikipedia.org/wiki/Artificial_intelligence.
[6] For further reading on ML please refer to https://en.wikipedia.org/wiki/Machine_learning

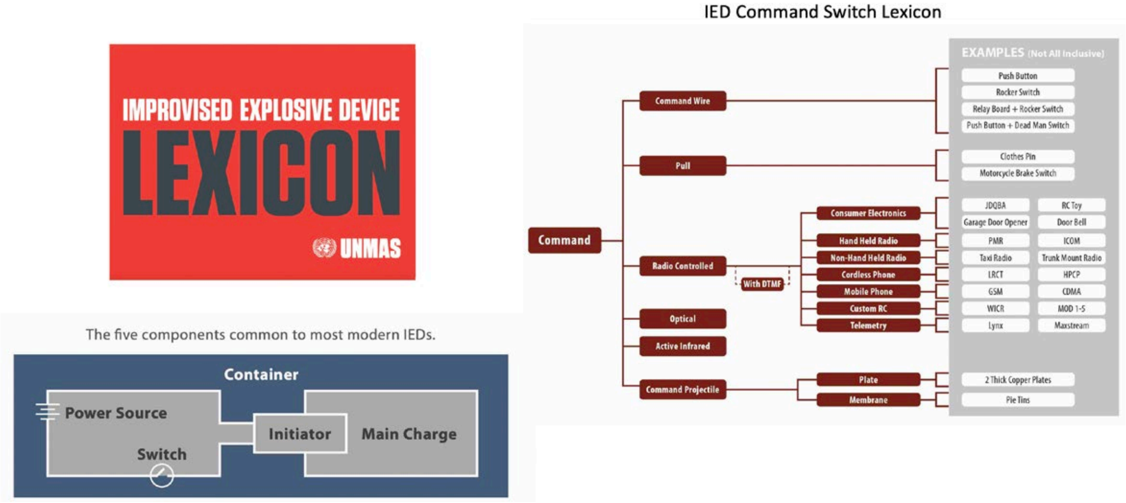
HCEP Discovery and Early Application

During Operation Enduring Freedom (OEF), in Afghanistan, and Operation Iraqi Freedom (OIF), the improvised explosive device (IED) posed a highly effective weapon against U.S and allied ground forces. An IED was typically constructed from basic materials that could be purchased at a hardware store. To mitigate this ongoing and pervasive threat, the DoD invested heavily in explosive ordnance disposal (EOD) technologies, such as robotics, ground penetrating radar, jammers, Mine-Resistant Ambush Protected (MRAP) vehicles, etc. Another key development out of this effort was the formalization of Weapons Technical Intelligence (WTI). WTI incorporated five basic mission types that fully addressed the warfighter Counter-IED intelligence needs. These missions included “force protection,” “signature characterization,” “sourcing,” “targeting,” and “support to prosecution.” During this time, Cogility was selected to help address this mission and to develop an analytic that could detect bomb signatures directly from sensors and bomb exploitation data and fuse that data and develop insights.

The project started at the Army Research Lab (ARL) where the HCEP analytic concept was first applied by analyzing bomb chemical signatures. After successfully demonstrating how HCEP could use chemical identifiers (e.g., ammonium nitrate) and match to known bomb chemical signatures, the project then moved to the Joint IED Defeat Organization (JIEDDO)[7] for further development. JIEDDO officials determined that this capability could best be used by warfighters in theater on what is known as the “edge.” This required Cogility to develop an application that would be easy to use (no code), be able to do analysis temporarily disconnected from networks, be database agnostic, and be lexicon driven. Concurrent to this, JIEDDO had been spearheading an effort to develop a WTI lexicon that provided precise definitions of bomb components to help U.S. and allies communicate consistently about IED exploitation and allow for tools to better search for IED related information.

Figure 1 shows a document that was created to provide a basis for WTI to inform the community and aid in developing tools and technology.

Figure 1. UNMAS Lexicon with extracts



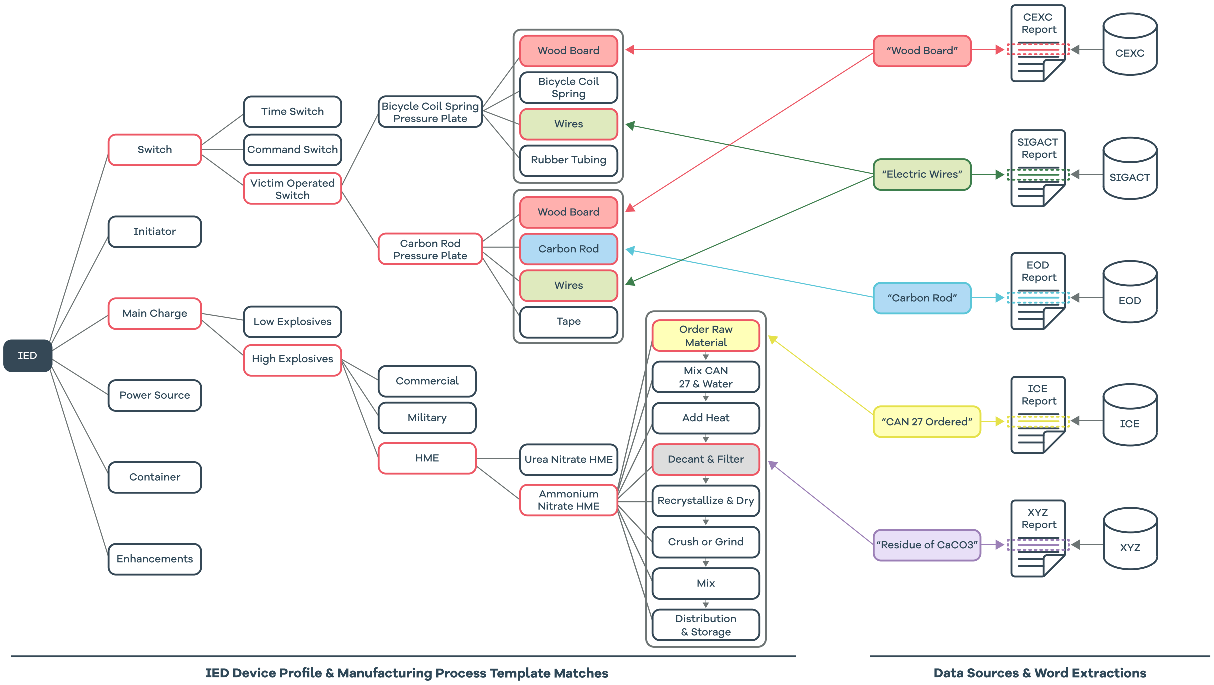
[7] https://cbrnecentral.com/profiles/name/joint-improvised-explosive-device-defeat-organization-jieddo/

In 2011, Cogility developed and demonstrated the first functional prototype with the ability to operate on a single laptop, ingest data from multiple IED data sources, process IED exploitation data, and identify IED signatures within a certain geographic area. Figure 2 depicts the hierarchical process that was used to match IED exploitation data to IED profiles. In this example, the application would query each IED exploitation database and scan the reporting and match words identified by the WTI lexicon,[8] and these “words” (i.e., Wood Board, Electric Wires, etc.) represent IED components that are matched to an IED profile (shown in the hierarchy). In the example, the colored icons represent matches. The bottom-up matching process represents the likely IED design in each area. In the example, the “Wood Board,” “Carbon Rod,” “Wires” likely matches a “Carbon Rod Pressure Plate,” “Victim Operated Switch” IED. This information is very useful to soldiers knowing that this type of IED could be in their patrol area, and the fact that Carbon Rods are a low ferrous material, thus mostly nullifying the benefit of metal detectors, and instead informs operators to use ground penetrating radar to better detect this type of IED. Also note, the “Carbon Rod Pressure Plate” profile is only partially completed, noting the “Tape” was not detected. This analytic approach infers the likely IED profile helping inform the human analyst and decision makers make more informed decisions.

> This information is very useful to soldiers knowing that this type of IED could be in their patrol area.

After this successful demonstration, the project was moved to Digital Ground Systems-Army (DCGS-A) and established as a program of record and named the Joint IED Analysis Tool (JIST) to further develop the application while working with analysts and subject matter experts.

Figure 2. HCEP C-IED Pattern Matching Process Example

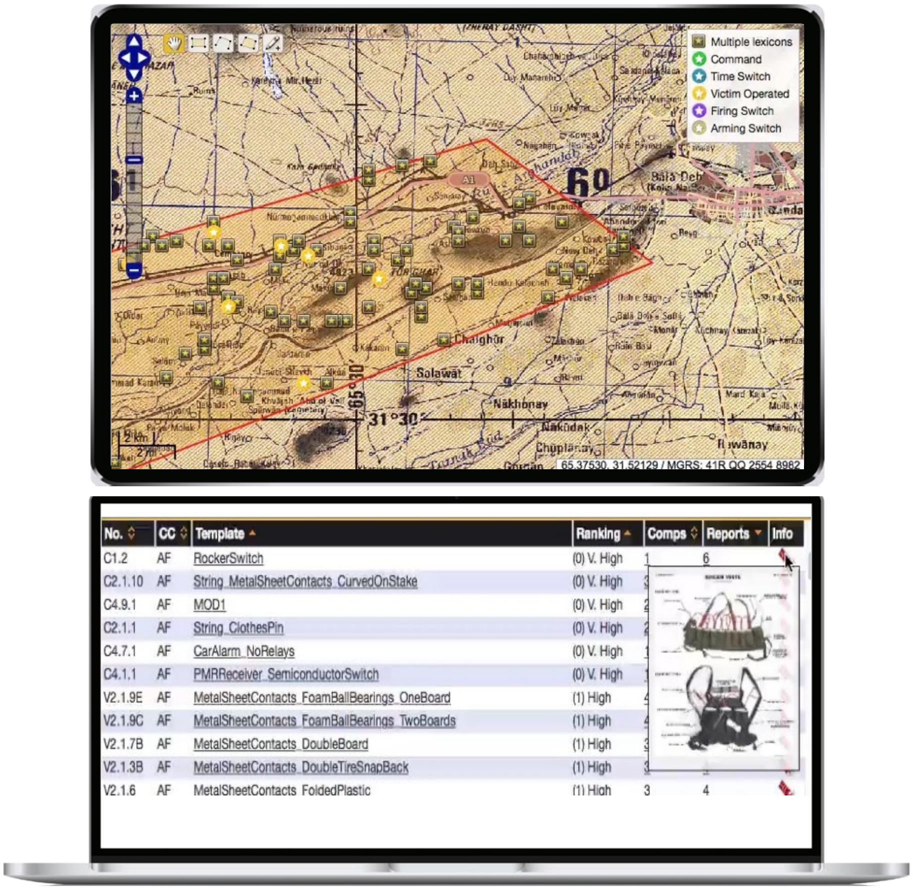


[8] https://unmas.org/sites/default/files/unmas_ied_lexicon_0.pdf

After several years and development cycles, JIST became the most advanced WTI platform ever used in the field by the U.S. Government. JIST was field-tested in OIF by analysts where it drastically improved the WTI analysis process. One of the biggest compliments that we ever received is from the Commander of Combined Joint Task Force Paladin,[9] who stated that JIST could be used by junior analysts to develop highly relevant intelligence in a very short period—within an hour—which would normally take a PhD level operations researcher days to produce.

Figure 3 shows 2 JIST visualizations exemplifying the type of analysis provided by JIST. The top image is a map display, and the bottom image is an IED profile analysis. Note, in this example, the data is synthetically created. In the map visualization, the red polygon represents the search area, and the map icons represent IED switch types discovered in the identified area. The map legend shows the types of lexicons that can be displayed (e.g., Multiple lexicons, Command, Time Switch, Victim Operated, and Arming Switch). The map also allows the user to drill down on the map where specific IED types are further revealed. The bottom view shows a prioritized list of likely IED profiles in the search area. In this example, it shows a "Rocker Switch" is the likely IED switch type to be employed in the search area and indicates this switch type is used in explosive vests.

Figure 3. JIST HCEP IED Geo and IED Profile Matches Visualizations



No.	CC	Template	Ranking	Comps	Reports	Info
C1.2	AF	RockerSwitch	(0) V. High	1	6	
C2.1.10	AF	String_MetalSheetContacts_CurvedOnStake	(0) V. High	3	[illegible]	
C4.9.1	AF	MOD1	(0) V. High	2	[illegible]	
C2.1.1	AF	String_ClothesPin	(0) V. High	2	[illegible]	
C4.7.1	AF	CarAlarm_NoRelays	(0) V. High	1	[illegible]	
C4.1.1	AF	PMRReceiver_SemiconductorSwitch	(0) V. High	1	[illegible]	
V2.1.9E	AF	MetalSheetContacts_FoamBallBearings_OneBoard	(1) High	4	[illegible]	
V2.1.9C	AF	MetalSheetContacts_FoamBallBearings_TwoBoards	(1) High	4	[illegible]	
V2.1.7B	AF	MetalSheetContacts_DoubleBoard	(1) High	3	[illegible]	
V2.1.3B	AF	MetalSheetContacts_DoubleTireSnapBack	(1) High	3	[illegible]	
V2.1.6	AF	MetalSheetContacts_FoldedPlastic	(1) High	3	4	

[9] https://en.wikipedia.org/wiki/Combined_Joint_Task_Force_Paladin

Since the development of JIST, Cogility continues to formalize HCEP and its application to many analytic problems in the intelligence field. The continuous evolution of analytic problems requires additional levels of adaptation and abstraction that allow analysts to effectively develop situational awareness of dynamic sociotechnical environments. In addition, emerging threats require new, highly agile, actionable intelligence products in such areas as supply chain risk, insider threat, information warfare, and cyber threat intelligence. HCEP continues to prove itself as a reliable foundation for a wide range of tailored intelligence products.

Prior to our discovery of HCEP and its implementation, Dr. David Luckham, Professor Emeritus at Stanford University,[10] was addressing similar analytic challenges in various commercial applications. Based on his CEP research, Dr. Luckham authored the book *Power of Events*[11] which provides a formal theoretical basis for CEP introducing the concept of abstraction hierarchies. Some early CEP implementations were applied to stock trading, where traders could define certain market conditions and automatically instigate a buy or sell order. While the focus of this paper is not to restate Dr. Luckham's concepts verbatim, our foundational concepts are very much aligned. We met with Dr. Luckham and did a side-by-side comparison of our two approaches. In many cases, where we saw explicit alignment in concepts, we adopted his terminology so customers could better relate to HCEP. Since Dr. Luckham's early work, there have been many significant technological advances to better enable HCEP, such as Apache Kafka[12] and Apache Flink[13], both streaming technologies which vastly improve scalability and performance over query-based systems. We continue to develop and enhance HCEP on a continuous basis. The foundation concepts will be further explained in the next section.

> Emerging threats require new, highly agile, actionable intelligence products in such areas as supply chain risk, insider threat, information warfare, and cyber threat intelligence.

[10] https://profiles.stanford.edu/david-luckham

[11] *The Power of Events, An Introduction to Complex Event Processing in Distributed Enterprise Systems,* David Luckham, 2002

[12] https://kafka.apache.org

[13] https://flink.apache.org/

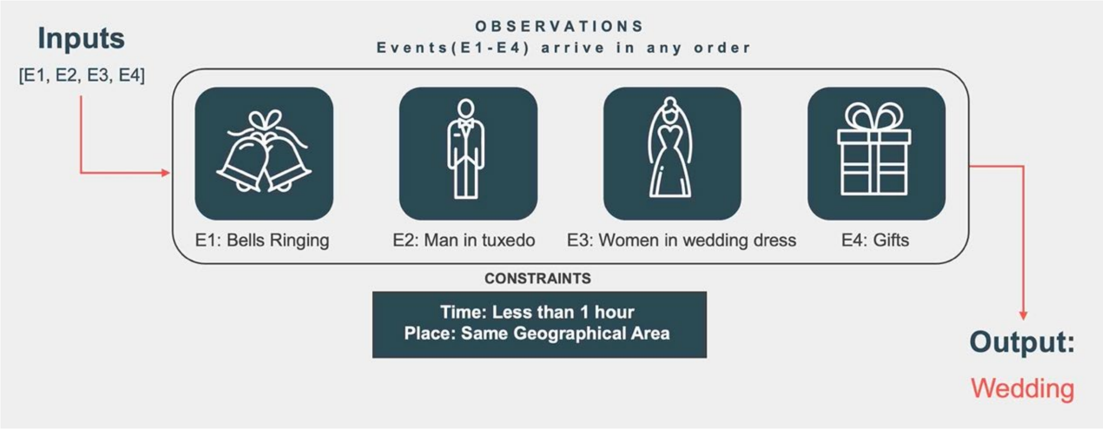
HCEP Foundation Concepts

As described in earlier paragraphs, HCEP was developed as the WTI application for analysts in the field where their lives depended on good intelligence. HCEP was discovered by applying practical problem-solving methods derived from computer science, software engineering, and systems thinking on how to discover patterns in data. The root of this is the ability to apply the powers of abstraction and decomposition. The ability to do multi-layer event processing, matching, and creation of new "abstract events" became the basis for HCEP.

Complex Event Processing Example–A wedding

As an example, shown in Figure 4, suppose someone wishes to know about all the weddings taking place at any given time in Los Angeles on any given day. For this example, we define a western wedding requiring observation of four events: "bells ringing," "man in a tuxedo," "woman in wedding dress," and "gifts," and by various means these events must be observed in roughly the same location and within one hour. To a computer this is just data, if a human observed these events, just two of the events: "man in tuxedo" and "woman in white dress," a human would infer this could be wedding, if the observations were made in separate locations, suppose more than a mile apart, or more than an hour apart, a human would quickly reject this inference.

Figure 4. Complex Event Processing Example



In this example the computer was able to match the events "man in tuxedo" and "woman in wedding dress" within the defined constraints, and later the events, "bells ringing," and "gifts" were observed within the 1-hour constraint. This is what we would call a fully "matched pattern." Based on this "matched pattern," in HCEP, we then "assert" that a wedding is occurring, meaning that we do not really know for sure, but it met our pattern requirements. If three of the four events were matched, then a human still may infer that a wedding is taking place because it's possible the bell was broken, or the gifts were hidden so the pattern may not completely be matched. However, just knowing this helps the human infer that a wedding is taking place. Later, we will discuss statistical inferences that can be an additional aid in determining the likelihood of partially matched patterns. Also note, in this example, we do not require the observed events to arrive in any order, which in many cases is true in the real world.

This wedding example exemplifies the types of analytic problems faced by analysts and decision-makers. The ability to discover patterns and infer the meaning of the pattern is essential in developing intelligence, allowing decision-makers to make informed decisions in time-sensitive situations.

HCEP Building Blocks

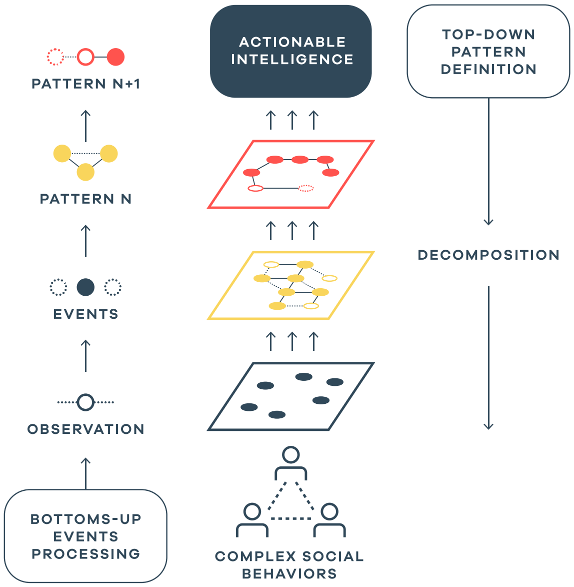
HCEP contains six foundational capabilities that allow us to detect patterns at scale and to be easily employed by non-technical users:

- Computation Hierarchy
- Stateful CEP
- Continuous Risk Assessment
- Lexicons
- Manual Actions
- Event Pattern Constraint Language

Computation Hierarchy

Computation Hierarchy considers both abstraction and decomposition as a conformal structure concept which forms the basis of HCEP. The HCEP hierarchy consists of event patterns organized in a hierarchical form. The hierarchy is typically conceived and developed from the top-down (see Figure 5) starting with a hypothesis to discover specific patterns in data that will answer the hypothesis. The HCEP structure is decomposed to a level that can match observed event data, which is referred to as the leaf or lowest level of the hierarchy. Data is processed from the bottom up where the matching process takes place. Matched patterns create new “complex events” that trigger event patterns at the next level and so on. The employment of computation hierarchical concept helps avoid combinatorial explosion. Combinatorial explosion can happen in rule-based systems that attempt to address every permutation of a rule. In HCEP, we can define multiple variants of a pattern at one level that each produce the same complex event. To illustrate this point, in the case of the wedding example discussed earlier, we can define many different versions of weddings, for example a Hindu wedding, or Jewish wedding will have different observables and therefore their own patterns, but the complex event is the same, a wedding event! This ability to manage complexity is key to HCEP and its ability to scale and handle extraordinarily complex analytic problems.

Figure 5. HCEP Computation Hierarchy



Stateful CEP

CEP is the process of matching events according to patterns, where each pattern defines explicitly what events constitute matching (e.g., string, structure field, etc.) and must satisfy one or more constraints with other potential matched events to generate a new or complex event. Each pattern is also stateful,[14] meaning that it will receive a matched event and retain that knowledge indefinitely, such as remembering the most recent bank transaction. As data is received, a partial pattern can be recognized, i.e., some of the events required by the pattern are found. The state of each pattern is maintained and updated as new matched events arrive. This updates the knowledge represented by a given pattern.

Figure 6. HCEP Event Pattern Example

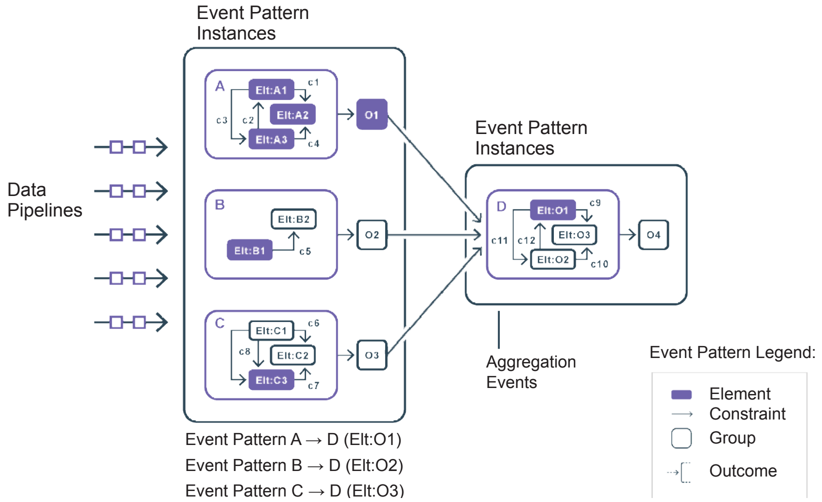


Figure 6 represents how HCEP works. In this example, there are four event patterns: A, B, C and D. The respective event patterns are shown receiving and filtering data (note the input data pipelines), and in the case of event pattern A, we can see that all three event types have been matched, as shown in Elt: A1, A2 and A3, and all the constraints (C1 through C4) have been satisfied. As a result, the event pattern has been fully satisfied and generates a new event O1, which becomes an input to event pattern D. Likewise, event patterns B, C, and D are partially matched and this set of event patterns will be static until a new event arrives that matches one of the non-matched event types (e.g., event pattern B, event type Elt: B2, and event pattern C, event types Elt: C1 and Elt: C2.) Note that elements in the pattern can, where appropriate, match multiple events of the same type (such as collecting bridesmaids at a wedding).

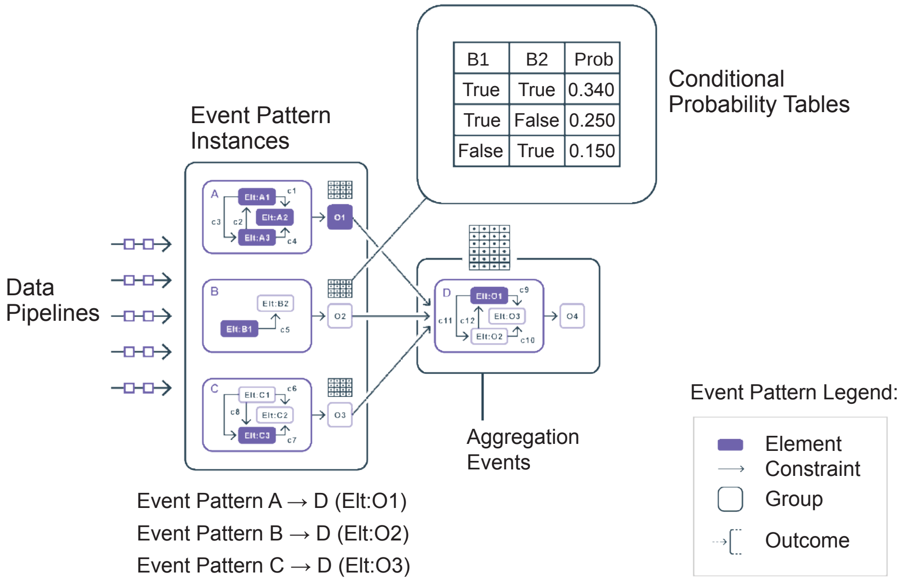
This example demonstrates several powerful concepts of causality between event relationships, something especially important when having to explain why your analytic came to a particular conclusion. In this example, the relationship between events shown in event pattern A, which shows Elt: A1 thru Elt: A3 generates O1. The causal relationship between Elt: A1 thru Elt: A3 show the causal relationship to event O1. This is also described as inferred causality.[15] The other concept is statefulness of the patterns that maintains the matched and partially matched event patterns. The "state" of these patterns represents our knowledge about the patterns we care about. This statefulness is something that analysts use to see trends and infer potential outcomes and therefore be more proactive in given situations.

[14] https://en.wikipedia.org/wiki/State_(computer_science)
[15] *The Power of Events, An Introduction to Complex Event Processing in Distributed Enterprise Systems*, David Luckham, 2002

Continuous Risk Assessment

In addition to being stateful, HCEP supplies users the option to employ risk assessment to determine the likelihood of events occurring in the future. Cogynt employs Bayesian inference for each event type and associates a conditional probability. These probabilities are subjective but reflect what the analyst or subject matter expert believes is important. HCEP applies Bayes rule[16] to perform the hierarchical calculation and updates its calculations accordingly as event types are matched thereby adjusting the calculation of the likelihood of the hierarchy of event patterns being completed. Figure 6 is the same event pattern example described in the paragraph above, except in this diagram, we show the assignment of conditional probabilities to each event pattern.

Figure 7. HCEP Event Pattern Example w/ Risk Assessment



B1	B2	Prob
True	True	0.340
True	False	0.250
False	True	0.150

In this example, event pattern A has completed and produces a factual O1, while event patterns B, C, and D are partially matched. In this example, Figure 7, showing the shaded event types (i.e., event pattern B: Elt B2, event pattern C: Elt: C1 and C2, and event pattern D: Elt O2 and O3). Instead of being static as described before, these patterns now generate one factual event O1, and two statistical events (O2 and O3) which are inputs to the higher-level pattern (event pattern, D). Event Pattern D processes these events generating event O4 as a statistical event. This example now supplies explicit statistical inference on the likelihood of a future event. The earlier example required humans to interpret the completed patterns and infer without the aid of statistical inference.

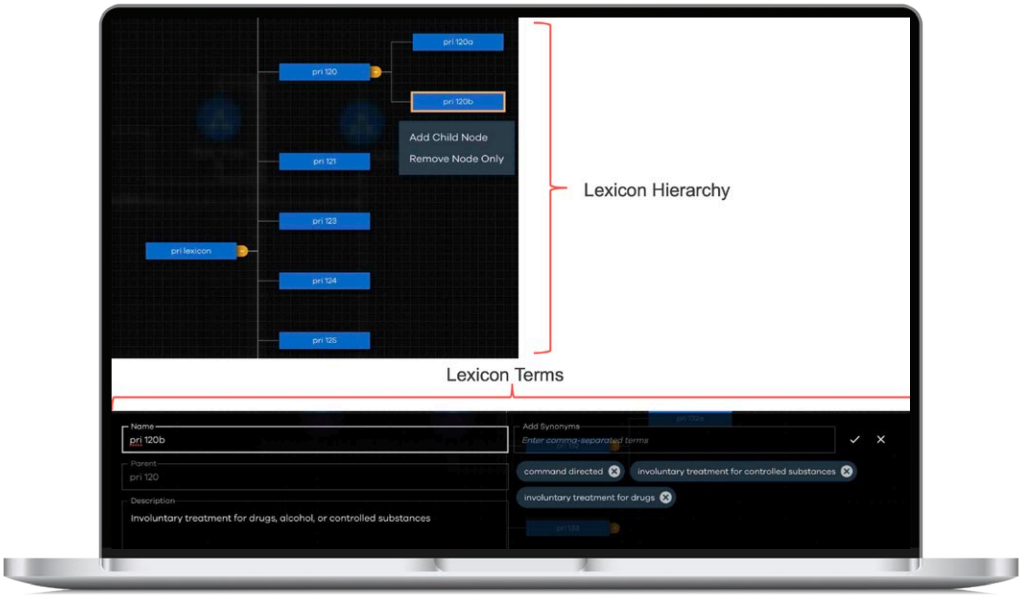
This continuous risk assessment gives us insights about developing situations based on risk or opportunity, which allows us to proactively mitigate risks or better leverage opportunities.

[16] https://en.wikipedia.org/wiki/Bayes%27_theorem

Lexicons

A lexicon is vocabulary of a language or branch of knowledge,[17] in this case, that of IEDs. For Cogynt and HCEP, we commonly analyze unstructured text such as reports, and the narratives in these reports define specific terms that analysts look for in extracting knowledge about what they are analyzing.

Figure 8. Lexicon Hierarchy Diagram



Name: pri 120b

Parent: pri 120

Description: Involuntary treatment for drugs, alcohol, or controlled substances

Add Synonyms: Enter comma-separated terms

command directed ⊗ involuntary treatment for controlled substances ⊗ involuntary treatment for drugs ⊗

Cogynt makes this easy by allowing the user to define a lexicon, which can be organized in the form of a tree (Figure 8) and decomposed as a hierarchy of terms. These terms, in some cases, serve as the matching relationship between the detected lexicon term and the event type, where we treat the matched word as an event. Cogynt also supports word variability such as tense, stemming, and wild card replacements to make lexicons more powerful. This is a powerful way to quickly extract complex narratives and pull the information most relevant to an analyst and relate this to patterns and behavior. An example of this being done is described earlier in the paper shown in Figure 2 and the extraction of IED exploitation reports and matching terms to IED device profiles.

Manual Actions

HCEP is a form of an expert system, and the models are derived from human knowledge. This means that, like any analytic, HCEP can get the wrong answer depending on the context. Manual actions are a means for the human to interact with the hierarchical processing and modify a matched event pattern. To address this challenge, HCEP allows for “manual actions.” Under certain circumstances, an analyst can generate an event for a specific instance of a pattern. This event will be input to the compute processing for that pattern which can reverse it, revise its risk level, or change the nature of the hypothesis matched. This result is propagated throughout the entire hierarchy and updates the risk scores. This is equivalent to an undo or other edit. If this situation becomes repetitive, then the model isn’t specific enough and will need modification. This is one feature and advantage that Cogynt offers, allowing for human judgement.

[17] https://en.wikipedia.org/wiki/Lexicon

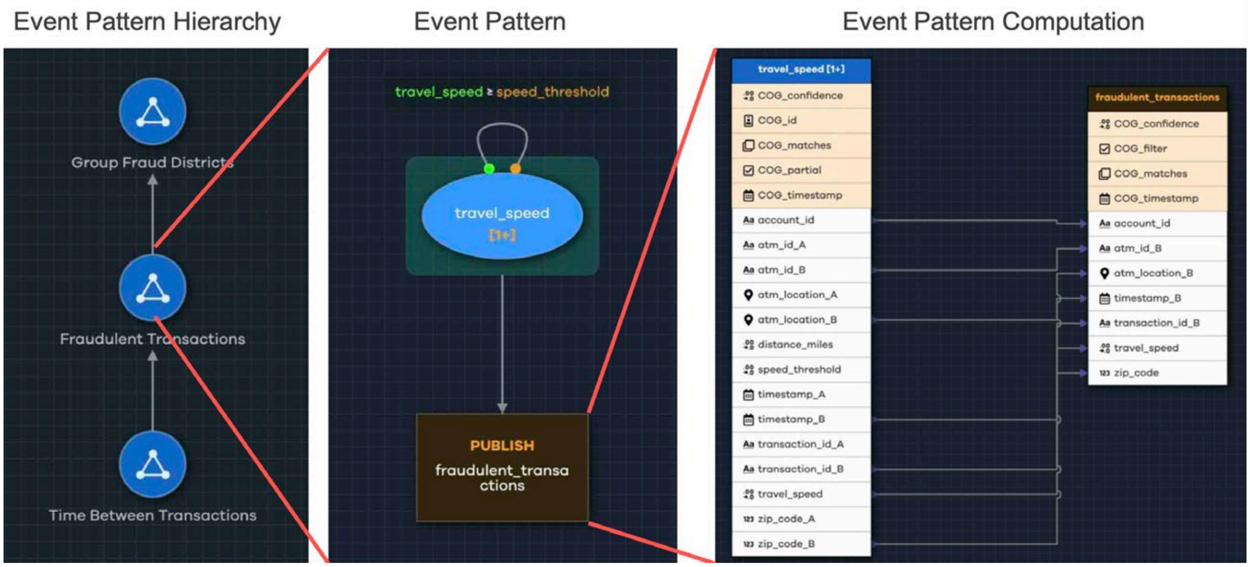
Event Pattern Constraint Language

The Event Pattern Constraint Language (EPCL) is a domain-specific language[18] that is declarative[19] graphical no-code programming language developed by Cogility. The Cogynt Authoring tool allows users to author models, discover schemas of input data sources (i.e., Kafka topics[20]), and create event pattern hierarchies. Another key aspect of EPCL and the Cogynt Authoring tool is computations. Computations allow for the mapping of data from input to output for matched event patterns performing mathematical calculations and/or transformations of the incoming data. The result of this process can be passed onto higher level event patterns, thus supplying context for the sociotechnical phenomenon under study.

Figure 9 illustrates Cogynt Authoring graphical user interface for authoring event patterns. Shown from left to right is an Event Pattern Hierarchy indicating the relationships between the event patterns. The center diagram is an Event Pattern with a single event type. The connecting line labeled “travel speed ≥ speed threshold” shows that the calculated travel speed is greater than or equal to a defined speed threshold. These specific values are defined in the Event Pattern Computation model shown on the far right, which is shown within the outcome event called “publish fraudulent transactions.” event. The Event Pattern Computation model shows the relationship of input to output data within the event pattern. These three modeling views form the basis for HCEP modeling within Cogynt. While the model shown in Figure 9 is a basic example meant to explain the concept, there is no limit to the depth and breadth of an Event Pattern Hierarchy.

> The Cogynt Authoring tool allows users to author models, discover schemas of input data sources and create event pattern hierarchies.

Figure 9. HCEP ECPL Graphical Constructs for an ATM Fraud Example

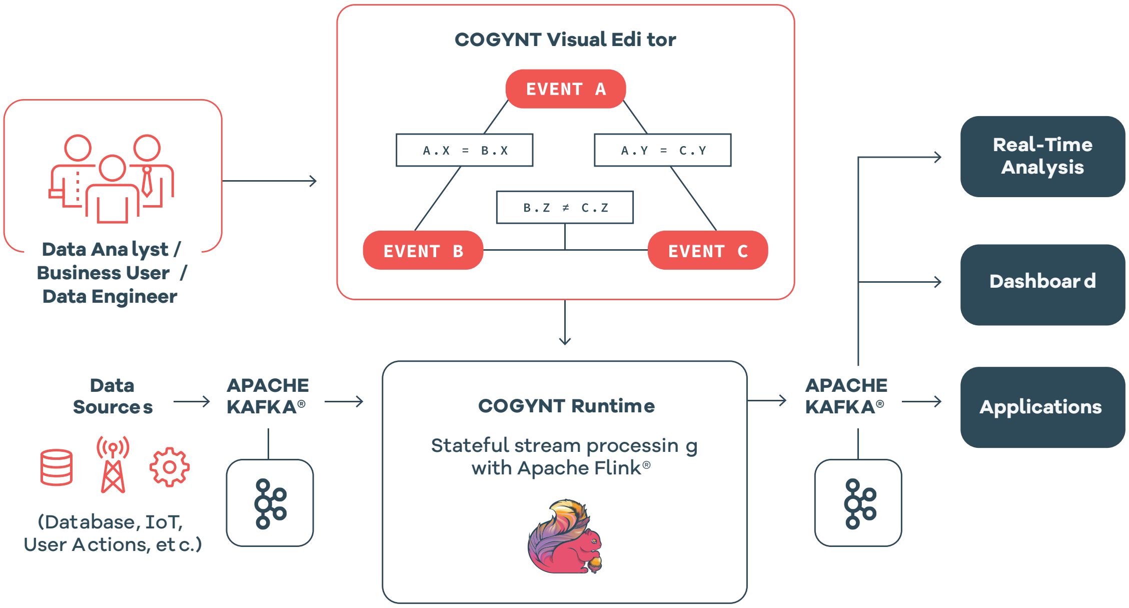


[18] https://en.wikipedia.org/wiki/Domain-specific_language
[19] https://en.wikipedia.org/wiki/Declarative_programming
[20] https://kafka.apache.org/08/documentation.html#introduction

Cogynt: HCEP Technology Enablers

Within the past five years, four technology enablers have allowed Cogility to develop HCEP in its current form. These technologies are Apache Kafka for stream processing, serving as a stream processing engine with stream storage; Apache Flink as a continuous real time and stateful compute engine; the cloud with Kubernetes enabling elastic scalability and management of a distributed microservice architecture; and real time streaming Business Intelligence tools like Apache Pinot[21] and Apache Superset.[22] Figure 10 is a high-level architecture of the Cogynt CI platform that shows the relationships between the key software components just described.

Figure 10. Cogynt CI Platform High Level Architecture



The Cogynt CI platform, and HCEP specifically, requires data analysts to create and deploy models to the Cogynt runtime (Apache Flink) and data to be streamed, or batched, to the platform from an external source to Apache Kafka. Apache Kafka then streams the data to Apache Flink where the live HCEP model runtime takes place. The results of the HCEP runtime are then streamed back into Apache Kafka for storage. Apache Flink may access the analytic results for further analysis if the event patterns require it. The results are available for external use or integration and tools such as the Analyst Workstation and Dashboards.

The use of Apache Kafka as a platform allows Cogynt to be open core architecture. The customer can view the analytic results in any application they choose. Cogility offers the Analyst Workstation as an option to support analysts with customized views such as a drill down view (Event Hierarchy) that shows the causal relationships between the events, which is described in greater detail later in this paper.

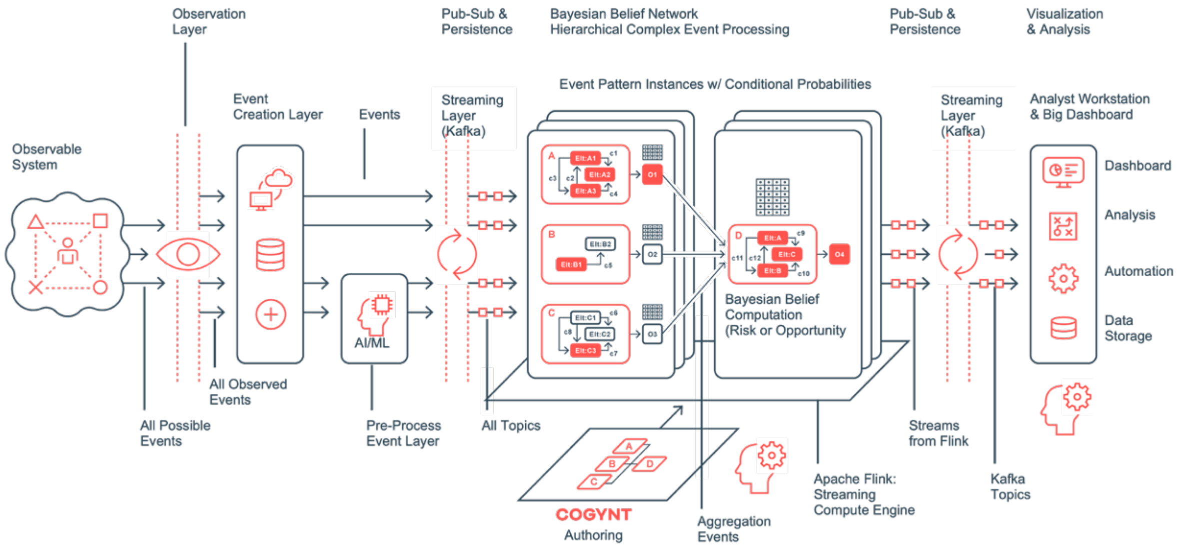
[21] https://pinot.apache.org/
[22] https://superset.apache.org/

Cogynt CI Platform and AI/ML Analytics Integration

Figure 11 illustrates the end-to-end data and analytics processing flow from the sourcing of events through processing of events, including HCEP, to display. One thing to note in this illustration is the application of AI/ML analytics in conjunction with Cogynt and HCEP. We have found that there are many scenarios where HCEP complements AI/ML and vice versa. Several common AI/ML analytics include facial recognition and speech transcription. These AI/ML analytics can be easily integrated as data and analytic pipelines that can feed Cogynt/HCEP analysis for complex behaviors and processes. This emphasizes the point that Cogynt is an open core platform and can complement many different analytic strategies adding immediate value to organizations looking to get better value of their data and existing analytic capabilities.

> Cogynt is an open core platform and can complement many different analytic strategies adding immediate value to organizations.

Figure 11. Cogynt CI Platform End-to-End Logical Architecture with HCEP



HCEP Visualizations–Behavioral Analytic Views

The Cogynt Analyst Workstation provides a wide assortment of analyst views for performing investigative tasks to validate HCEP generated notifications to determine if a given risk or opportunity requires validation. HCEP produces two essential views that help analysts validate behavioral risk or opportunity (Figures 12 and 13). Figure 12 shows risk over time, with each data point representing an event. The analyst has the means of inspecting all the points on the graph for validation

Figure 12. Temporal Risk Assessment Graph

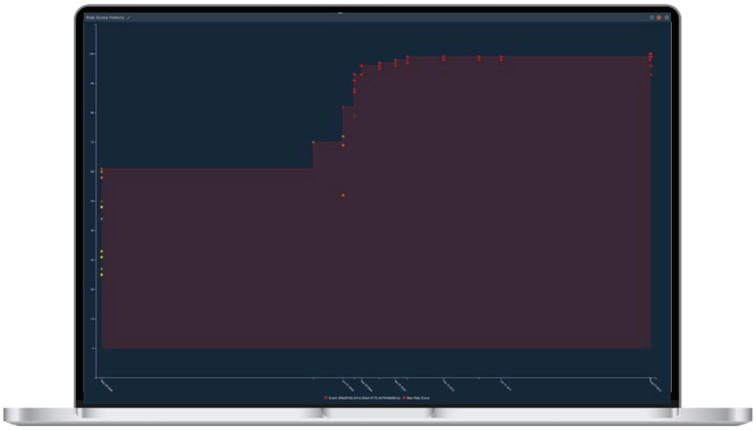
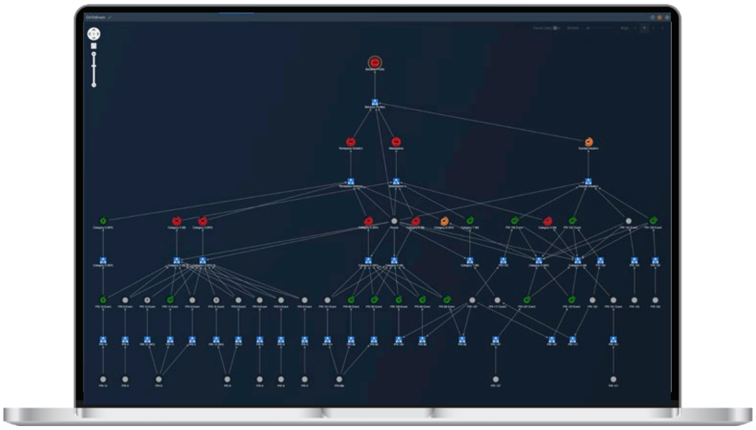


Figure 13 is an event hierarchy that shows the hierarchical relationships between the events and event patterns and associated risk. The analyst can also inspect any of the events in this event hierarchy for validation.

Figure 13. Event Hierarchy Drill Down View



In summary, these views show the causal relationship between events and what the events mean in terms of risk or opportunity. These explicit views make HCEP “explainable”—which is critical to gain trust in the analytic solution where consequential decisions are aided by these types of analytic views.

Conclusion

HCEP was originally developed to meet the need to recognize IED bomb signatures quickly and accurately, to better inform soldiers on the battlefield, and save lives. Through this process, Cogility recognized that use cases of this type occur in every sector, along with the need to discover patterns in data and translate data into information. This theme has driven Cogility to develop Cogynt and employ the latest open-source streaming technologies. We can now deploy Cogynt and HCEP to address the largest and most difficult problems in areas of cyber security, supply chain, insider threat, medical, and many other sociotechnical risk—or opportunity—assessment challenges.