

Incorporating Semantic Knowledge into Dynamic Data Processing for Smart Power Grids

Qunzhi Zhou, Yogesh Simmhan, and Viktor Prasanna

University of Southern California, Los Angeles, USA
{qunzhizh, simmhan, prasanna}@usc.edu

Abstract. Semantic Web allows us to model and query time-invariant or slowly evolving knowledge using ontologies. Emerging applications in Cyber Physical Systems such as Smart Power Grids that require continuous information monitoring and integration present novel opportunities and challenges for Semantic Web technologies. Semantic Web is promising to model diverse Smart Grid domain knowledge for enhanced situation awareness and response by multi-disciplinary participants. However, current technology does pose a performance overhead for dynamic analysis of sensor measurements. In this paper, we combine semantic web and complex event processing for stream based semantic querying. We illustrate its adoption in the USC Campus Micro-Grid for detecting and enacting dynamic response strategies to peak power situations by diverse user roles. We also describe the semantic ontology and event query model that supports this. Further, we introduce and evaluate caching techniques to improve the response time for semantic event queries to meet our application needs and enable sustainable energy management.

Keywords: Semantic Web, complex event processing, smart grid

1 Introduction

The power grid is undergoing rapid modernization into a Smart Grid through the integration of digital and information technologies. This trend is worldwide [1]. Conventional meters which record the accumulative power usage at monthly base are being replaced by smart meters which report power consumption at minutes interval. In Europe, for example, Italy and Sweden are approaching 100 percent deployment of smart meters for consumers. In U.S., the largest municipal utility, the Los Angeles Department of Water and Power (LADWP), has begun to expand its advanced metering infrastructure. At the building level, ambient sensors and smart appliances, such as HVAC sensors, occupancy sensors and plug-in electric vehicles (PEVs) are being integrated into building control systems. These various information sources provide fine-grained monitoring and control capability of power use activities, in both spatial and temporal scales. However, transforming this capability to actionable knowledge is challenging, due to the complexity of both information and interactions.

Demand response (DR) is a cornerstone application in Smart Grids that aims to curtail power load during peak load periods. This reduces the captive power

generation required by a utility for reliable operations by shaping the power usage to remain relatively constant over time. Existing DR approaches are statically planned through time-based pricing incentives for voluntary curtailment by customers. But the intermittent nature of renewable generation like wind power and changes in energy use pattern hampers this static approach.

Dynamic DR [18] supplements traditional DR approaches by leveraging real-time information for online decision making, enabling opportunistic curtailment based on the current situation. However, as an application layer software, a dynamic DR system has to deal with the heterogeneous and constantly evolving Smart Grid infrastructure. Further, the curtailment decision and enactment is distributed, engaging participants like utility operators, facility managers, building occupants and household customers. No single person has a holistic sense of the information space and concepts used for DR decisions.

Semantic Web provides an ontology-based extensible framework that allows information to be shared and reused across application and domain boundaries. It has been used for information integration in domains such as health care [10, 20], biology [21, 16] and transportation [11]. However these approaches mostly process time-invariant or slowly evolving semantic data.

We provide a framework that adopts semantic knowledge in stream processing and applies it to dynamic DR in Smart Grids, specifically the *USC Campus Micro-Grid*. We combine complex event processing (CEP) with Semantic Web to facilitate high level user application design. The semantically enhanced CEP system ties in with our semantic repository that hosts comprehensive information on the campus micro-grid. These are intended to be leveraged by the campus energy center, building managers, staff and students, and their applications, for campus energy management. Semantic technologies are central to meet the organic growth of information and infrastructure diversity and keep them accessible for easy use. Our key contribution in this paper are:

Semantics in Use in Micro-Grid. We discuss specific uses and benefits of semantic technologies for micro Smart Grid applications and the participants.

Semantics for Complex Event Processing. We describe unique benefits offered by semantics for complex event processing.

Semantic CEP (SCEP) Optimizations. We introduce caching techniques for efficient semantic query processing over event data streams, and evaluate them using semantic CEP queries used for dynamic DR on campus.

The rest of the paper introduces dynamic DR and semantic information model for the campus Micro Grid (§ 2), describes the uses of semantic technologies in the Micro Grid (§ 3), presents our semantic CEP model and DR patterns (§ 4), discusses and evaluates our caching optimizations for query processing (§ 5), reviews related work (§ 6) and presents our conclusions (§ 7).

2 Background

Our work is done as part of the Los Angeles Smart Grid Demonstration Project, where the USC Campus is a *Micro-Grid* testbed for evaluating Smart Grid

technologies and software tools. Smart Grids have two characteristics relevant to Semantic Web: Diversity and Evolution. Smart Grid applications need to support **diverse information sources and users**. Besides sensors and instruments monitoring the infrastructure to produce an avalanche of data, information on electrical equipment, organizations, class schedules and weather are also used for decision making [22]. The managers and consumers of this data also vary. At USC, the facilities management services (FMS) deploys and manages sensors and meters in campus buildings for energy monitoring and control. Research and service groups such as the sustainability center, energy club, and energy forecasting models consume this data for analysis. Users like FMS operators, department coordinators and students need an integrated and easy to use view of the complex data to support their individual needs.

Another feature of Smart Grids is their **continuous evolution**, given the emerging nature of technologies and deployment. For e.g., USC, as the largest private power consumer in Los Angeles, has over 60,000 students, faculty and staff spread over 170 buildings [23]. This means that infrastructure is constantly being upgraded and consumers change every year. Of late, an average of two new buildings are built each year on campus, each with hundreds of sensors and equipment. Ambient sensors such as temperature, airflow and CO₂ sensors are deployed at the room-level to monitor conditions. Likewise, around 19,000 new students enroll in USC each year which induces changes in power usage profiles in dormitories and classrooms. Smart Grid applications need *sustainably adapt* to these changes in the information space with low overhead.

2.1 Online Strategies for Dynamic Demand Response

Traditional DR approaches are static: the decision is global and made in advance [1]. Dynamic DR as introduced in [18] supplements traditional DR by offering a more fine-grained approach that is responsive to dynamic power usage changes. The pervasive sensing capability enables us to monitor power consumption and its indirect influencers, such as weather and occupancy, in near real-time. Dynamic DR adopts a data driven approach that detects the occurrence of specific information patterns by examining hundreds or thousands of online data streams. Such analysis offers deeper situational awareness on power usage behavior for timely and opportunistic curtailment strategies.

Our objective is to provide a framework where multi-disciplinary users can *define* DR situations at higher level abstractions, and *detect* these situations over dynamic Smart Grid data streams for timely decision making. Sample dynamic DR situations are listed below, with more details in [18].

Situation 1. The *space temperature* in an *office* of *EE department* is lower than the *green building temperature*.

Situation 2. The *power use* of a *teaching building* exceeds its *pre-peak demand*.

Situation 3. The *space temperature* in a non-occupied *meeting room* is lower than the *green building temperature*.

Situation 4. *Fan coils* in building *MHP* peak concurrently.

Situation 5. The *temperature* in a *meeting room* is above setpoint by 5 °F.

2.2 Semantic Information Model for Smart Grid Applications

Earlier, we have designed an integrated, modular Smart Grid semantic information model for dynamic DR applications [24, 22]. In summary, this captures, **Data Sources.** We model Smart Grid data sources and the information they emit, including smart meters which measure the power use of buildings, sensors which detect room occupancy, temperature and airflow sensors which measure HVAC status, weather reporting services. These sources are linked to concepts of physical and virtual spaces where they monitor.

Infrastructure. We model both the campus power grid infrastructure, such as the distribution network, and the physical environment. This includes concepts and relationships between building, rooms, and energy sinks like appliances and equipment. These relate to power usage behaviors and curtailment capabilities. For example, a *meeting room* may need to be cooled only when occupied.

Organization. Campus organizations including schools, departments, laboratories and so on are also modeled. These can help users to define organization-specific DR strategies even though it may span physical locations. For example, a monitoring pattern can alert the department coordinator when consumption exceeds a certain threshold.

Other Information. Other information including scheduling and weather also help DR applications, and are modeled using existing domain ontologies.

3 Semantics in Action on Campus Micro Grid

The semantic Smart Grid information model forms the center piece of many applications in the USC campus Micro Grid. These range from asset management, information diffusion and data analysis, allowing different participants to cooperate on campus-wide DR operations. In the following we discuss these applications and their use of semantics.

3.1 Sustainable Asset Management

Asset management is a basic power grid operation. In a Smart Grid, asset management must shift from a dependence on domain experts' experiences to a reusable knowledge base. This is necessary to deal with an aging (but experienced) workforce whose retirement can lead to less experienced staff without holistic knowledge of the evolving grid. Another concern is the need for seamless and rapid integration of new resources deployed in the power grid.

As a Smart Grid testbed, the USC campus Micro Grid is pioneering novel infrastructure and its management. The USC FMS deploys and manages assets on campus, replacing electromechanical devices by digital instruments, introducing power efficient equipment and enabling bi-directional grid communications. At the time of writing, smart meters were deployed to monitor over 170 buildings on campus, and over 50,000 sensors installed to monitor room temperature, HVAC airflow and fan speed, and even CO₂ levels. As equipments are upgraded, ensuring transfer of knowledge and its accessibility to relevant users is a challenge.

Semantic technologies enable sustainable asset management for the campus Micro Grid. We worked with USC FMS to build the Smart Grid ontology model which captures relevant Micro Grid aspects ranging from electrical equipments, buildings, participants, and departments [24]. Using this model, asset management merely involves performing model queries that are synchronized with field operations. For example, when a new occupancy sensor is installed, the facility operator inserts the sensor entity into the semantic repository, describing it and its relations to existing domain entities using properties such as “ee:hasID” and “ee:hasLocation”. A new sensor type can be intuitively introduced by creating a concept such as “ee:OccupancySensor” in the model, defining its properties and classifying it under an existing parent category such as “ee:Sensor”. Compared to relational model, semantic ontologies support property inheritance and reasoning while ensuring that introduction of new concepts, or in other word enriching the schema, does not affect legacy data. Further, external models (such as Weather) can be easily integrated. Semantic model based asset management hence ensures rapid and extensible knowledge transference and integration.

3.2 Accessible Information Diffusion

Another key Smart Grid activity is to promote energy awareness and participation by delivering interpretable energy use information to power consumers and end-use applications. We use a web portal as the primary vehicle for this task. Incorporating semantics in these applications considerably improves the process of information dissemination in a heterogeneous power grid environment. It provides ubiquitous data and query representation that hides complexities associated with multi-disciplinary users and distributed asset management.

Our prototype web portal for campus Micro Grid information exploration, *eScope*, is hosted at smartgrid.usc.edu. It provides not only static asset information but also dynamic energy use “heatmap” for the campus. The portal supports the needs of both domain experts for easy exploration and end consumers wishing to learn about the energy footprint. It uses SPARQL queries to extract and present this information from a semantic repository; we use *4store*. A sophisticated information integration pipeline running on a private Eucalyptus Cloud continuously retrieves raw data tuples from various information sources including sensors, maps them into RDF triples using rules, and inserts them into the semantic database where it is linked with domain ontologies.

Using semantics allows easy information retrieval by even non-experts, such as portal developers. This is particularly important in a multi-disciplinary project. For example, the USC Micro Grid does not follow a consistent sensor naming scheme. The same type of sensor type, such as kilowatt-hour sensor, has opaque names like “D163Watts”, “XLP0100103000022UD” or “BIE_TotalWatts” for different buildings, making it challenging for the portal to maintain static queries for display results as a heatmap. By using semantic concepts rather than a relational database, the portal developer can query for conceptual terms using SPARQL, such as recent “ee:Kilowatt” readings from all “bd:TeachingBuilding”:

```

SELECT ?building, ?time, ?kwhReading
WHERE {
  ?event evt:hasSource ?src . ?src ee:hasLocation ?loc .
  ?loc rdf:type bd:TeachingBuilding . ?loc bd:hasCode ?building .
  ?event rdf:type ee:Kilowatt . ?event evt:hasValue ?kwhReading .
  ?event evt:hasTimestamp ?time . FILTER(?time > "10:15:00")}
ORDER BY DESC(?time)

```

This also allows new portal features to be incorporated *rapidly* by just understanding a few concept terms, and *seamlessly* reflects infrastructure upgrades.

3.3 Data Analysis

Data analytics for both off-line demand forecast as well as online DR optimizations utilize the semantic knowledge.

Adaptive Planning using Machine Learned Forecasting. We use machine learning to train power consumption forecasting models using historical energy use data [3] that are useful for planning equipment upgrades, maintenance schedules, and curtailment policies. These models use different semantic features as indirect influencers of energy use, such as the types of buildings, rooms and customers, academic schedules, weather conditions, and so on. However, not all features may be relevant for prediction at all times. The web portal allows a data analyst to easily explore the semantic knowledge base to identify candidate features of interest. Historical values of these identified features, along with the power consumption of that target buildings, are then extracted from the semantic repository and normalized into a form that can be consumed by a Regression Tree training model. Semantics allow the analyst – who is not an expert on power systems – to still navigate the domain models and pick potential influences of energy use, allowing knowledge to be easily imparted without requiring a domain expert by her side.

Online Optimization using Semantic CEP. Online data analytics for DR uses complex event processing (CEP) for detecting real-time situations, represented as event patterns, from among streams of events. The limitation of current CEP systems in processing only structural patterns impedes their effective use in an information rich domain like Smart Grid. Existing systems process events streaming from sensors as plain relational data tuples. As such, complex event patterns can only be defined as a combination of attributes presented in event data. Users have to know the details of event structures and sources before defining low level pattern specifications.

We have introduced semantics into CEP [25] as a solution to meet online DR requirements. Details of *SCEPter*, our semantic CEP system, is outside the scope of this work. However, it does offer several advantages.

Interoperability. The broad space of software and hardware vendors in Smart Grids means that different standards need to co-exist. This extends to data formats and schemas. For example, airflow sensors on campus use different variants of the “airflow” attribute such as “flowrate” and “airvolume” in their event

format. With a traditional CEP system, pattern designers are exposed to the structural heterogeneity of events and have to rewrite the same query for different data streams whose formats may vary. A Semantic CEP system helps capture these distinctions, for example using “owl:sameAs” relations, which allows a unified *conceptual* query specification over heterogeneous event formats without in-depth knowledge of standards. This also reduces the complexity operational debugging by having a smaller set of conceptual patterns.

Expressivity. Traditional CEP systems process events solely based on the attributes they possess in the event tuple. By mapping events and their tuple attributes to as part of the semantic ontology, query constraints can then be defined on related domain concepts and entities. This significantly enhances the power of an event pattern specification in detecting very precise situations, while eliminating false positives.

Accessibility. Defining DR event patterns over domain ontologies shield users from lower level details of data streams and their changes. As shown in the examples in § 2, we can easily define patterns that apply to only *meeting rooms* on campus, even if the user has no idea of which buildings have meeting rooms, let alone the sensors that are deployed in those rooms.

In the following sections, we discuss the semantic-enriched event processing approach and optimizations for Smart Grid applications in detail.

4 Semantic Complex Event Processing Model

We provide a semantic stream query language and a *data-driven* processing engine for dynamic DR applications in Smart Grid [25]. Data access systems are typically data-driven or query-driven [5] based on what initiates/completes the operation. CEP systems are data-driven as a pattern is detected when the last event required for a complete match arrives. Query-driven systems such as relational/semantic databases (often) evaluate results as soon as a query is submitted. Conceptually, our system is data-driven as incoming events continuously trigger pattern evaluation. However, since our semantic CEP pattern is specified over both static data in semantic repositories and dynamic stream data, the implementation causes the *event-triggered pattern evaluation* to incorporate both a semantic query part and a subsequent CEP pattern detection.

4.1 Semantic Event Model

The state-of-the-art CEP systems [2, 13] process primitive events as relational data tuples, i.e.,

$$primitive\ event ::= \langle attributes; timestamp \rangle$$

Based on the relational model, complex events (event patterns) are defined as compositions of primitive events with attribute constraints. Directly applying CEP systems for dynamic DR requires users define DR event patterns at data

level and synchronize patterns with the grid infrastructure upgrades. For example, data schema of space temperature and occupancy measurement streams on USC campus is,

$$\text{event tuple} = \langle \text{sensorID}, \text{reading}; \text{timestamp} \rangle$$

To define Situation 1 as a traditional CEP pattern, users have to explicitly specify the list of thermostat which locate in offices of EE department and keep it up-to-date in the query. In addition, as CEP patterns can only be matched by evaluating syntactic identical attributes, semantic mismatches between user vocabularies and event data have to be addressed manually.

To overcome these limitations, we propose to link dynamic data streams with background ontologies to process semantics of events. Figure 1 shows an example of semantic temperature measurement event. It's essentially a RDF event graph connected to domain ontologies with properties materialized from the original data tuple.

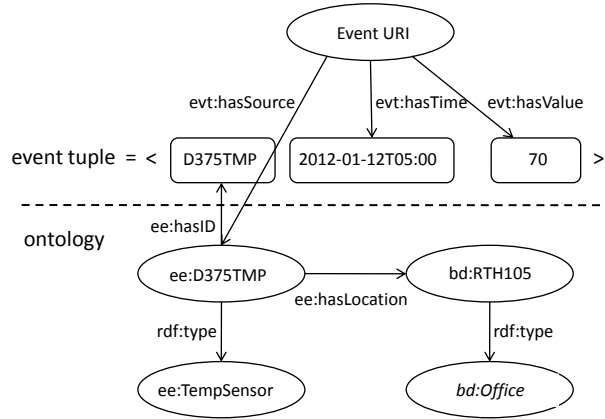


Fig. 1: Semantic Event Linked with Background Knowledge

4.2 Semantic Event Query Model

We propose a two-segment query model over the semantic events described above for dynamic DR situation modeling. The general query structure is,

$$\begin{aligned} \text{Semantic Event Pattern} ::= & \\ & [\text{semantic filtering subpattern}]\{0, n\} \\ & [\text{syntactic CEP subpattern}]^+ \end{aligned}$$

A semantic event pattern in our system consists of two types of subpatterns:

semantic filtering subpatterns and syntactic CEP subpatterns. The semantic filtering subpatterns are SPARQL queries to specify semantic constraints and annotations of input events. A pattern can have 0 to n semantic subpatterns where n is the number of streams the pattern correlate. The syntactic CEP subpatterns are traditional CEP queries which specify temporal and logic constraints over filtered and enriched event tuples. One pattern can have 0 or 1 CEP subpattern. For example, Situation 1 can be modeled as a pattern with 1 semantic filtering subpattern and 0 CEP subpattern. The semantic filtering subpattern in SPARQL is,

```
SELECT ?e
WHERE {
    ?e evt:hasSource ?src .
    ?src ee:hasLocation ?loc .
    ?loc rdf:type bd:Office .
    ?loc bd:belongs org:EE.Department .
    ?src rdf:type ee:TempSensor .
    ?e evt:hasValue ?reading .
    bd:GreenBuildingTemp bd:hasValue ?val .
    FILTER(?reading < ?val) }
```

Consider another example, the pattern for Situation 3 correlates 2 streams including the occupancy and temperature measurement streams. It has a semantic filtering subpattern on each stream to query/select event location and constrain the location type as “bd:MeetingRoom”. The semantic subpattern for the temperature measurement stream also specifies the temperature reading is less than “bd:GreenBuildingTemp”. Denote “?o” as filtered events from the occupancy stream and “?t” as filtered events from the temperature stream and assume the filtered event tuples are annotated with a new location attribute “loc”, the CEP subpattern for Situation 3 represented in Siddhi [19] is,

```
AND (?t, ?o)
CONDITION {(?t.loc = ?o.loc) and (?o.reading = false)}
```

We developed the semantic event query processing system around an existing CEP engine kernel, Siddhi. When a new event tuple arrives on input streams, the corresponding semantic events are materialized and combined with domain ontologies to evaluate semantic subpatterns. Qualified data tuples are extended and passed to the CEP engine for syntactic subpattern matching. For details of the processing engine please refer to [25].

5 Cache Optimization for Continuous Querying

In this section, we discuss optimizations for processing semantic filtering subpatterns over event streams. The baseline approach is to perform semantic queries

whenever a new event arrives. However, semantic querying is known as time expensive. It performs inference and self-join operations over the ontology knowledge base. In general, a semantic query with a single path expression requires $(n-1)$ self-joins over the ontology where n is length of the path.

We developed caching algorithms to improve the performance of semantic stream querying. As an initial effort, we make the following assumptions,

- The semantic filtering queries do not correlate multiple events.
- The semantic filtering queries do not contain alternative or disjunctive triple patterns.

5.1 Query Caching

The key observation for query caching is multiple events may share semantic query results so that the system need not evaluate queries for all events. A semantic event is modeled as a directed tree whose root node is the event URI and has edges linked to event properties as shown in Figure 1. We introduce the following definition,

Definition 1. The *event root properties* of a semantic event are the properties directly materialized from its data tuple attributes.

As examples, for the semantic event shown in Figure 1, “ee:D375TMP”, “2012-01-12T05:00” and “70” are *event root properties*.

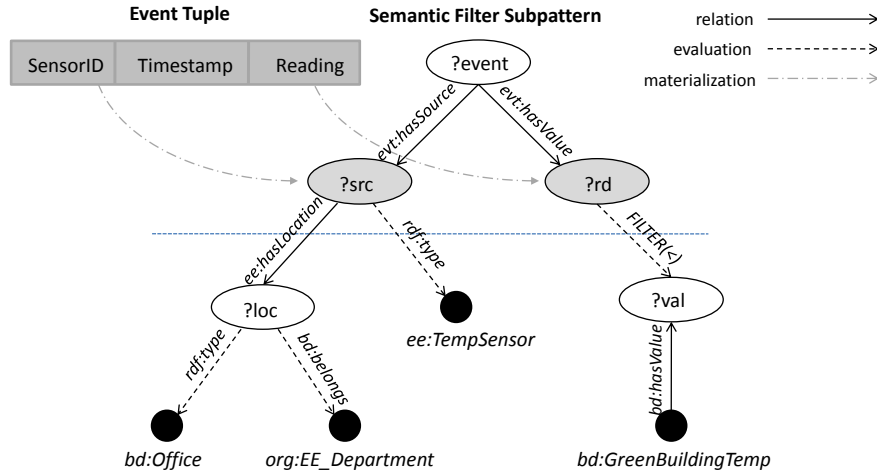


Fig. 2: Semantic Event Query Graph

On the other hand, a semantic event query can also be modeled as a directed tree whose root node is an event variable and it has edges connected to property variables, literals, ontology classes or instances. Executing a semantic event query

Algorithm 1 Semantic Stream Query with Query Caching

Require: Cache table ht initialized for Q , domain ontologies O **Ensure:** Evaluation result v for query Q

```

1: while Receiving semantic event  $e$  do
2:   Compute cache key  $k$  for  $e$ 
3:    $v \leftarrow ht.get(k)$ 
4:   if  $v! = null$  then
5:     Return  $v$ 
6:   else
7:      $v \leftarrow Evaluate(Q, e, O)$ 
8:     Update  $\{k, v\}$  to  $ht$  based on LFU policy
9:     Return  $v$ 
10:  end if
11: end while

```

is essentially finding event trees that match the query tree. As an example, Figure 2 shows the query tree for the semantic filtering subpattern of Situation 1.

As shown in the query tree, the inner nodes are all variables and leaf nodes are either literals, ontology classes or instances. The edges can be classified as relation or evaluation edges. We further define,

Definition 2. The *query root properties* of a semantic event for a query are the event root properties which are evaluated in the query.

For example, for the semantic event shown in Figure 1, “ee:D375TMP” and “70” are *query root properties* for the query tree shown in Figure 2. For any query which satisfies the assumptions stated before, its tree graph can be decomposed as conjunctive paths from the event variable node, through the variable nodes of *query root properties* to leaf nodes. Whether a semantic query is evaluated to true or false for an event is hence completely determined by the *query root properties* of the event for that query. In other word, if two events share the same *query root properties* for a query, they can share the query result.

Based on the above observation, we design the query caching mechanism as,

Cache data structure. The cache is implemented as a collection of hash tables. The SCEP system initializes and maintains one hash table for each query. The *query root properties* are used as the cache key and the boolean query evaluation result is the cache value.

Cache lookup/update When a new event e arrives, the system fetches the *query root properties* of e for a query Q and look up the corresponding hash table for matches. If it hit the cache, we use the cached result without actually performing the query. Otherwise the system materializes the semantic event for querying and updates the cache. Currently, we implemented a simple Least-Frequently-Used (LFU) update strategy.

Denote the boolean function that evaluates query Q over event e and domain ontologies O as $Evaluate(Q, e, O)$, the pseudo code for semantic event query with query caching is shown in Algorithm 1.

5.2 Path Caching

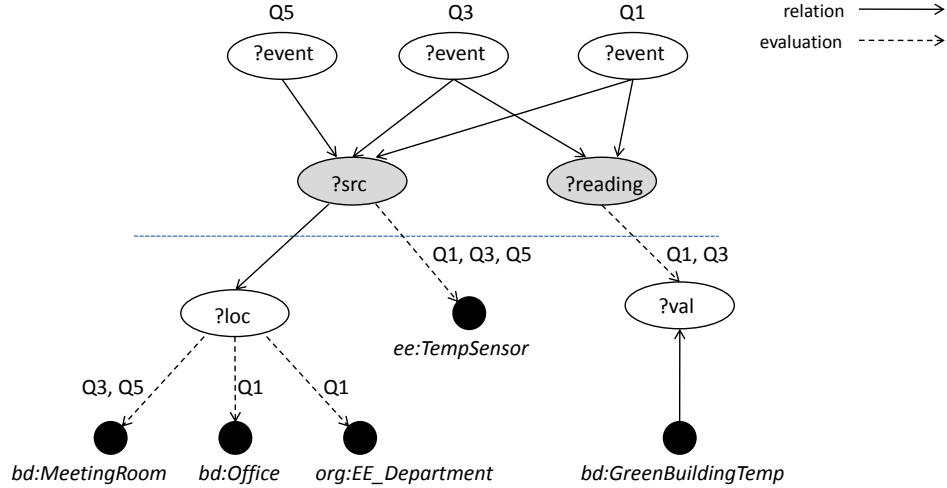


Fig. 3: Sharing Paths between Semantic Event Queries

In addition to sharing query results between events, it also makes sense to reuse path evaluations between queries especially when a number of queries share a smaller set of path expressions. Consider Situation 1, 3 and 5, Figure 3 shows the path sharing between their semantic filtering subpatterns Q1, Q3 and Q5. It also should be noticed that path expressions with leaf nodes such as “bd:Office” and “bd:MeetingRoom”, which are semantically disjoint, only need to be evaluated once for events with the same query root properties. Based on above observations, in the second cache optimization approach we maintain cache tables for individual query path and update caches by referencing the semantic relations between paths. The pseudo code for semantic stream query with path caching is shown in Algorithm 2.

5.3 Evaluations

Experiments are conducted to evaluate the semantic caching algorithms in dynamic DR scenarios expected for the USC Campus Micro Grid. In these experiments, we run the SCEP system on a 12-core AMD Opteron server, with 2.8GHz cores, 32GB physical memory and running Windows Server 2008.

Data collected from HVAC systems and smart meters on the USC campus is used as experimental data streams. We performed two sets of experiments each for three times and the average values are reported here. In the first set of experiments, we submit 9 semantic CEP queries to the SCEP engine and compare the throughput of the system in the case without caching optimization,

Algorithm 2 Semantic Stream Query with Path Caching

Require: Cache table ht_i initialized for path P_i ($i = 1$ to n) of query Q , dht_{ij} ($j = 1$ to m) identified as disjoint cache tables of P_i , domain ontologies O

Ensure: Evaluation result v for query Q

- 1: **while** Receiving semantic event e **do**
- 2: Compute cache keys k_i of e for P_i ($i = 1$ to n)
- 3: $v \leftarrow true$
- 4: **for** $i = 1$ to n **do**
- 5: $v_i \leftarrow ht_i.get(k_i)$
- 6: **if** $v_i = true$ **then**
- 7: Continue
- 8: **else if** $v_i = false$ **then**
- 9: $v \leftarrow false$
- 10: Return v
- 11: **else**
- 12: $v_i \leftarrow Evaluate(P_i, e, O)$
- 13: Update $\{k_i, v_i\}$ to ht_i based on LFU policy
- 14: Update $\{k_i, !v_i\}$ to dht_{ij} ($j = 1$ to m) based on LFU policy
- 15: **end if**
- 16: **end for**
- 17: Return v
- 18: **end while**

with query caching and with path caching. In the second set of experiments we submit 120 queries to the engine and evaluate the time performances of the three algorithms again. Figure 4 shows the experiment results. Obviously, the number of queries has significant impact on the system performance. We expect to sample campus sensor data at 1-minute interval for the dynamic DR applications. This requires a minimum throughput of 83 events/second to handle around 5000 data points on campus. Without caching optimization, the system can merely process around 10 events per second with 9 queries and 0.7 event with 120 queries. The throughput of the system ranges from 130 to 2400 events/second in the two experiments with query caching and path caching. We also notice when the number of pattern increases, path caching usually outperforms query caching as it allows evaluation results to be shared between queries.

6 Related Work

Semantic Smart Grid Information Modeling. The power systems industry has been opaque, dominated by a few large companies with proprietary information stacks. Smart Grids are forcing this to change. Standards designed by organizations like IEC and NIST provide common protocols and data models that can be used by the various participants. There has also been recent work on developing semantic-level Smart Grid information integration framework. [4] proposed a shared ontology model to provide common semantics for Smart Grid applications. The ontology captures domain concepts by transforming existing

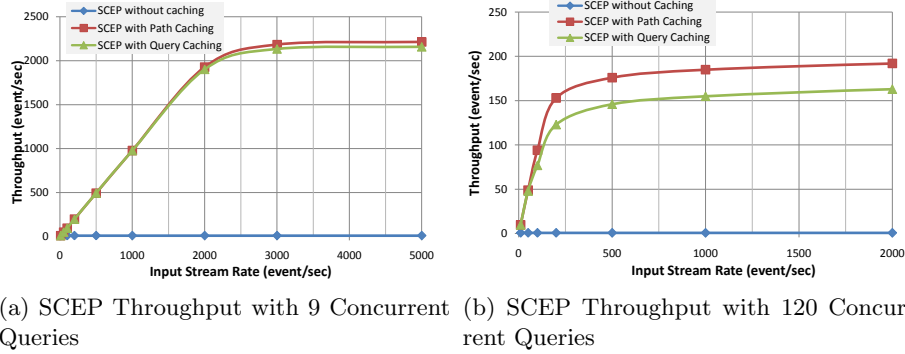


Fig. 4: Evaluation of Semantic Caching for Streaming SCEP Queries.

standards, such as IEC’s Common Information Model (CIM), to a uniform conceptual model. Our semantic Smart Grid information model can be considered as an extension to the model proposed in [4], and complements it with broader knowledge that is required for DR decision making. Besides power grid domain elements, we also link these with modular ontologies on physical spaces, organization, and weather that are crucial to DR applications.

Complex Event Processing Traditional CEP approaches like Cayuga [2] and SASE [13, 9] have focused on specifying and detecting temporal and logical relations among syntactical events modeled as an infinite sequences of relational tuples with interval-based timestamps. These use a SQL like query model with operators such as selection, projection, and conditional sequence.

The problem of semantic stream processing has been discussed in C-SPARQL [7] and ETALIS [5, 6]. C-SPARQL extends the SPARQL language with window and aggregation clauses to support RDF stream processing. However, while C-SPARQL extensively considers aggregation operations, it does not support several stream processing operators that are essential to Smart Grids, including temporal sequence and negation. ETALIS is a rule-based deductive system that acts as a unified execution engine for temporal pattern matching and semantic reasoning. It implements two languages for specification of event patterns: ETALIS Language for Events (ELE), and EP-SPARQL for stream reasoning. Both event patterns and semantic background knowledge are transformed to Prolog rules and executed by a Prolog inference engine for reasoning and pattern detection. However, these languages independently are not insufficient for our use cases. The ELE pattern language lacks semantic operators while EP-SPARQL supports few temporal operators such as sequence and optional sequence. Rather than adopt a bespoke solution that departs from traditional CEP systems, our proposed semantic CEP framework is a hybrid that leverages the native features of both CEP (Siddhi) and SPARQL engines to offer a richer query syntax. More practically, it also allowed for rapid construction of such a framework for our Micro Grid using existing tooling, and improves the performance using the proposed optimizations.

Semantic Caching. Semantic caching has been widely studied for database query optimization [14] by storing the results of previously queries locally. [12] discusses caching theory in terms of deciding when answers are in cache, and semantic overlap. [17] describes the use of semantic cache in an ontology-based web mediator system and considers extracting partial results from caches for new queries. A special feature of their approach is organizing cache by concepts and exploiting domain knowledge for defining queries to complement partial cache results. While not breaking new ground in caching strategies, we do apply it to a novel scenario of Semantic CEP where query performance over continuous event data can be punitive otherwise. Our caching algorithms for stream queries leverage existing state-of-the-art, resembling the global caching in [8, 12], and partial query caching [15]. Other than caching query results for subsequent relational queries, we cache semantic query results for new data.

7 Conclusion and Future Work

We have discussed incorporating semantics into Smart Grid applications and dynamic stream processing for the USC Micro Grid. Semantic Web domain ontologies form the foundation for diverse participants and DR applications to manage and access data conceptually. Realtime grid observations abstracted as semantic events allow intuitive definition and detection of semantic CEP patterns. Out caching optimizations improve its performance, as validated empirically.

Our work lies in two directions. First, we plan to extend the ontology models from campus Micro Grid to a utility scale, and identify additional semantic event patterns for DR strategies. Second, we will investigate additional optimizations to overcome performance bottlenecks of semantic event processing, which currently limit throughput to less than 3000 events/second even with cache optimization.

Acknowledgments. This work is supported by the Department of Energy under Award Number DE-OE0000192 and the Los Angeles Department of Water and Power. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, the LA DWP, nor any of their employees.

References

1. FERC assessment of demand response and advanced metering. Staff Report, December 2008.
2. A. Demers, J. Gehrke and et al. Cayuga: A general purpose event monitoring system. In *the Conference on Innovative Data Systems Research (CIDR)*, 2007.
3. A. Saima, Y. Simmhan and V. Prasanna. Improving energy use forecast for campus micro-grids using indirect indicators. In *International Workshop on Domain Driven Data Mining*, 2011.

4. J. L. Andrew Crapo, Xiaofeng Wang and R. Larson. The semantically enabled smart grid. Technical report.
5. D. Anicic, P. Fodor, R. Stuhmer, and N. Stojanovic. Event-driven approach for logic-based complex event processing. In *International Conference on Computational Science and Engineering*, 2009.
6. D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic. Stream reasoning and complex event processing in etalis. *Semantic Web Journal*, 2012.
7. B. Francesco, B. Daniele and et al. An execution environment for c-sparql queries. In *International Conference on Extending Database Technology (EDBT)*, 2010.
8. S. Dar, M. Franklin, B. Johnsson, D. Srivastava, and M. Tan. Semantic Data Cache and Replacement. In *Very Large Data Base Conference (VLDB)*, 1996.
9. Y. Diao, N. Immerman, and D. Gyllstrom. SASE+: An agile language for Kleene closure over event streams. Technical report, UMass, 2007.
10. T. Q. Dung and W. Kameyama. A proposal of ontology-based health care information extraction system: Vnhies. In *IEEE International Conference on Research, Innovation and Vision for the Future*, 2007.
11. E. D. Valle, I. Celino and D. Dell'Aglio. The experience of realizing a semantic web urban computing application. In *the Terra Cognita Workshop*, 2009.
12. P. Godfrey and J. Gryz. Answering queries by semantic caches. In *Database and Expert Systems Applications*. Springer, 1998.
13. D. Gyllstrom, E. Wu, and et. al. SASE: Complex event processing over streams. In *the 3rd Biennial Conference on Innovative Data Systems Research*, 2007.
14. A. Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 2001.
15. A. M. Keller and J. Basu. A predicate-based caching scheme for client-server database architectures. *The VLDB Journal*, 5, 1996.
16. P. W. Lord, S. Bechhofer, M. D. Wilkinson, and et al. Applying Semantic Web Services to Bioinformatics: Experiences Gained, Lessons Learnt. In *International Semantic Web Conference*, 2004.
17. M. Karnstedt, K. Sattler, I. Geist and et al. Semantic caching in ontology-based mediator systems. In *Berliner XML Tage*, 2003.
18. Q. Zhou, Y. Simmhan and V. Prasanna. On using semantic complex event processing for dynamic demand response optimization. Technical report, Computer Science Department, University of Southern California, 2012.
19. S. Suhothayan, K. Gajasinghe, I. Loku Narangoda, S. Chaturanga, S. Perera, and V. Nanayakkara. Siddhi: A second look at complex event processing architectures. In *ACM GCE Workshop*, 2011. <http://siddhi.sourceforge.net>.
20. T. Cui, S. Harold and et al. Time-oriented question answering from clinical narratives sing semantic-web techniques. In *Intl. Semantic Web Conference*, 2010.
21. C. Taswell. Doors to the semantic web and grid with a portal for biomedical computing. *IEEE Transactions on Information Technology in Biomedicine*, 2008.
22. Y. Simmhan, Q. Zhou and V. Prasanna. Chapter: Semantic Information Integration for Smart Grid Applications. 2011.
23. Y. Simmhan, S. Aman et al. An informatics approach to demand response optimization in Smart Grids. Technical report, USC, 2011.
24. Q. Zhou, S. Natarajan, Y. Simmhan, and V. Prasanna. Semantic information modeling for emerging applications in smart grid. In *IEEE Conference on Information Technology: New Generations*, 2012.
25. Q. Zhou, Y. Simmhan, and V. Prasanna. SCEPter: Semantic complex event processing over end-to-end data flows. Technical Report 12-926, Computer Science Department, University of Southern California, 2012.