Semantic Reasoning in Event Management System Using Ontology

Safeer Ahmed
Faculty of Computing
Ulster University
Birmingham, West Midlands
ahmed-S56@ulste.ac.uk

Abstract*— This report, authored by Mr. Safeer Ahmed, describes semantic reasoning in the Event Management System Ontology—an ontology framework built using OWL that addresses modeling challenges related to roles, resources, event scheduling, and their interrelationships. The ontology contains basic entities like Event, Attendee, Vendor, Venue, OrganizationalRole and their relationships, for example Wedding is a subclass of Party. Important object properties like hasVenue, attendsEvent, performsAt also allow integration of logistics while data properties like eventDate and hasSalary also add domain specific value. Combining class hierarchies with property restrictions and support for Description Logic (DL) queries, the ontology is capable of automated reasoning to derive useful but covered information such as event type classification, role allocation and provide efficient query solutions for tracking attendees or coordinating vendors. It provides solutions to data integration problems in computing environment, role definition, overlapping roles and rescheduling surprise events and other changing conditions, thus providing rich semantics and reusability which is beyond the reach of typical databases. The implementation shows a better inference capabilities of the knowledge structure and can be used in academic research and industry level event management systems.

***Keywords*— ontology, event management, knowledge representation, semantic web, OWL**

I. Introduction and Problem Description

Inefficiencies still exist in modern semantic event management system which system still rely on nonintegrated data stored in emails, spreadsheets and documents. Such ineffectiveness manifests in various slips like the aforementioned venue double-booking (GrandHallVenue), role confusion (Caterer vs. Decorator), and attendee mix-up (VIP_Guests vs. General_Attendees). Manually tracking events such as Wedding, Conference, or Concert, along with other manual processes, increases the delays, operational risks, and client dissatisfaction which need to be addressed with a cohesive strategy to consolidate and standardize event data.

An attempt to solve these problems is the introduction of an Semantic Reasoning Event Management System Ontology, constructed in Web Ontology Language (OWL). This encompasses core concepts such as Event (with subclasses Wedding and Conference), Attendee, Vendor along with Venue, and constrains them using object properties (hasVenue, providesCaterer) and data properties (eventDate). Automated conflict detection and advanced semantic queries such as "The DL query retrieves individuals who are Performers at both ICPOLH Conference and ABsWedding?" are made possible by the ontology. It replaces disparate systems with a unified system, which ensures flexibility with varying events while also meeting planning reliability demands for academia or industrial use.

Ontologies, which are formal representations of a set of concepts within a domain and the relations between those concepts, have been much instrumental ontologies in this regard. They have improved greatly data interoperability plus complex query operations that can be applied in any context where precise data synthesis and retrieval are needed.[2][3]

Earlier studies proved that applying ontologies in event management systems led to efficient operation plus boosted involvement as well as accuracy of data. The present report builds the ontology model for our proposed semantic reasoning model using Protégé.[4][5].

II. Justification About Ontology and Knowledge Representation and Reasoning

Knowledge representation approaches such as propositional and first-order logic are not suited for intricate fields such as event management with deep hierarchies. Propositional logic has no variables which makes it possible to only work with true/false statements, and although first-order logic works with quantifiers and variables, it fails at capturing relationships and semantic constraints that are hierarchical. Prolog, which is built on top of first-order logic, faces challenges related to scaling and semantic interoperability for dynamic systems despite offering rule-based reasoning. Unlike Prolog, ontologies created in OWL (Web Ontology Language) using Protégé provide a rich semantics framework. For example, OWL allows representation of classes like Event and Vendor and their properties such as hasVenue and providesCaterer, which is beyond the scope of Prolog's first-order logic.

This ontology goes beyond these frameworks by including detailed staff attributes like hasSalary for FinanceManager and metadata for vendors like Decorator and SoundSystemProvider that have advanced roles enabled by owl domain-specific constraints and axioms. Protégé's tools further facilitate advanced reasoning, automating conflict detection (e.g., venue double-booking) and inferring implicit knowledge (e.g., mandatory speakers for conferences). Unlike Prolog's static rules, OWL supports Description Logic (DL) querying across interconnected entities (e.g., retrieving events at GrandHallVenue), ensuring data integrity through axiomatic rules. By transforming fragmented data into an interoperable knowledge graph, this ontology addresses the limitations of less expressive systems, offering a scalable solution for semantic Event Management System.

Ontological representations have also been applied in various domains, healthcare being one of them. Here, ontologies allow systematic representations which can be

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

instrumental for the management of clinical data [6][7]. The ontology-based approach assures interoperability at much higher levels and enhanced reasoning leading in turn to better outcomes especially under complex decision making scenarios [6]. These would drastically minimize ambiguity in event management and therefore bring possible automated reasoning mechanisms for events which would enhance operational effectiveness.

III. PRIOR KNOWLEDGE REQUIRED ABOUT SEMANTIC REASONING IN EVENT MANAGEMENT ONTOLOGY

To effectively use the event management ontology, users must understand the following key areas:

A. *Semantic Reasoning in Event Management System Basics*

Knowledge of core event types (weddings, conferences, concerts), participant roles (attendees, speakers, performers), vendor services (caterers, decorators, photographers), and venue types (indoor/outdoor).[8][9]

B. *Ontology Structure*

Understanding how events are categorized (e.g., "Conference" is a type of "Event"), how relationships are defined (e.g., "hasSpeaker" connects events to speakers), and rules like "every conference must have a speaker.

C. *Data and Properties*

Familiarity with event details like dates, prices, and locations stored in the system, and how these connect to participants and venues.

D. *Software Tools*

Basic skills in using Protégé to view event relationships and DL to ask questions like "Which events use GrandHallVenue? [10][11]

For beginners, guides like Strategic planning kit for dummies [1]simplify these concepts.

E. *Software Tools*

One must have adequate knowledge regarding the specific context in which the ontology will be applied in event management to ensure the proper design of a knowledge representation model. [12][13]

F. *Reasoning Mechanisms*

The ability to deduce new knowledge from ontology by means of a reasoning engine serves as a mature but critical part when it comes to using ontologies in an effective manner within complex scenarios.

IV. METHODOLOGY AND IMPLEMENTATION

The ontology was systematically developed in Protégé using a structured approach to model Event Management Systems. The methodology encompasses the definition of six core class hierarchies, properties, and individuals, ensuring logical consistency and practical applicability.

A. *Concepts and Concept Hierarchies*

The ontology "Semantic Reasoning in Event Management System" is organized to depict the basic elements and relations that come into play when planning, handling, and executing an event so that smart reasoning and meaningful understanding can be facilitated concerning all aspects related to events. At its top level, this ontology is divided into six basic classes which correspond to the major realms within an event management system. These high-level classes form the semantic backbone for the whole ontology and are designed with extensibility in mind through well-defined subclass hierarchies.

Top-Level Classes

1) Attendee- An individual or group who takes part in the event.

2) Event- The central concept around which different forms of organized activities revolve.

3) Organizational Role - Defines different roles accountable for planning and execution.

4) Schedule - Captures time-related aspects of event planning.

5) Vendor- All service providers who render services towards facilitating the event.

6) Venue- This will show the types of places where events are held.

This hierarchy lays the groundwork for higher-level reasoning activities that allow for in-depth analysis, classification, and automation within the processes related to semantic reasoning in event management system. The Fig. 1. below depicts the highest-level class hierarchy of the ontology 'Semantic Reasoning in Event Management System', created using OWL structure. [15][16]

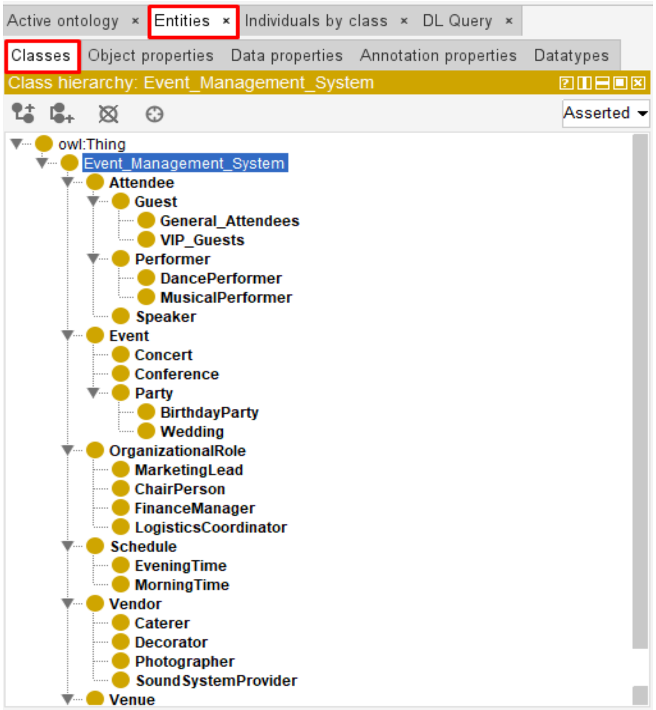

Fig. 1. Event Management System Ontology Class Hierarchy

Attendee Class and Its Subclasses

The Attendee class represents an individual or entity that is present or participating in an event. It is further divided into different sub-roles which represent the aspects that an attendee might typically embody at an occasion.

Subclasses of Attendee:

- Guest
  - GeneralAttendees
  - VIP_Guests
- Performer
  - DancePerformer
  - MusicalPerformer
- Speaker

This hierarchical design enables a better categorization of attendees based on their function and access level at the event.

Fig. 2. Screenshot below showing the Attendee class and its subclass hierarchy.

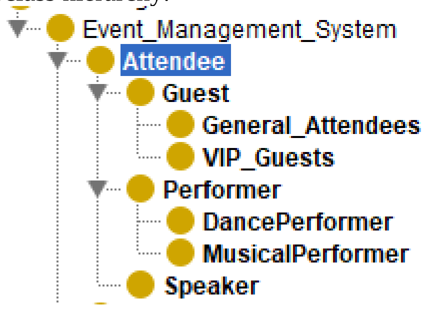

Fig. 2. Attendee Class and Its Subclasses

Event Class and Its Subclasses

The Event class represents the central activity being organized. It is further specialized into specific event types. This structure allows semantic differentiation and reasoning based on the nature of the event.

Subclasses of Event:

- Concert
- Conference
- Party
  - BirthdayParty
  - Wedding

This structure helps with detailed planning for various events. It's great for organizing things like logistics, guests, and services.

Fig. 3. Screenshot showing the Event class and its subclass hierarchy.

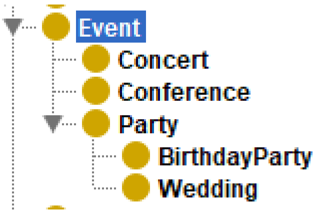

Fig. 3. Event Class and Its Subclasses

OrganizationalRole Class and Its Subclasses

The OrganizationalRole class models the roles taken by individuals or teams in managing the event.

Subclasses of OrganizationalRole:

- ChairPerson
- FinanceManager
- LogisticsCoordinator
- MarketingLead

By modeling these roles, the ontology supports role-based task assignment and responsibility tracking.

Fig. 4. Screenshot below showing the OrganizationalRole class and its subclass hierarchy.

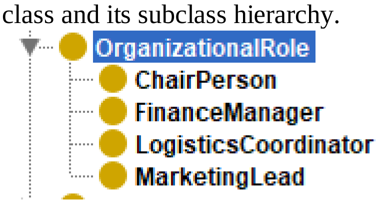

Fig. 4. OrganizationalRole Class and Its Subclasses

Schedule Class and Its Subclasses

The Schedule class captures the time-related aspects of events. It enables semantic structuring of time slots and periods.

Subclasses of Schedule:

- MorningTime
- EveningTime

Example Instance: MorningTimeABsWedding

This structure allows reasoning about availability, scheduling conflicts, and time-specific tasks.

Fig. 5. Screenshot showing the Schedule class and its subclass hierarchy.

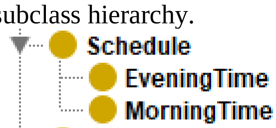

Fig. 5. Schedule Class and Its Subclasses

Vendor Class and Its Subclasses

The Vendor class captures service providers offering different types of support for events. Each subclass identifies a specific service type.

Subclasses of Vendor

- **Caterer**
  - *Example: EliteCatering*
- **Decorator**
  - *Example: MoonDecorator*
- **Photographer**
- **SoundSystemProvider**

This division allows organizers to semantically associate event needs with relevant providers.

Fig. 6. Screenshot showing the Vendor class and its subclass hierarchy.

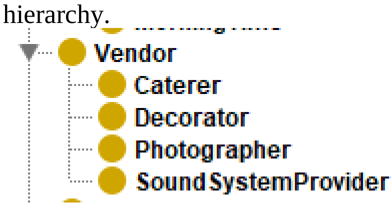

Fig. 6. Vendor Class and Its Subclasses

Venue Class and Its Subclasses

The Venue class defines the physical space where events are held. It is divided based on environment type.

Subclasses of Venue:

Indoor

Example: GrandHallVenue

Outdoor

Example: SparkalPark

This classification helps with event planning based on location type, availability, and capacity.

Fig. 6. Screenshot showing the Venue class and its subclass hierarchy.

Fig. 7. Venue Class and Its Subclasses

B. Relations (Properties) and Relation Hierarchies

Ontology employs properties to establish semantic relationships between entities. Object properties define interactions between classes, while data properties capture attributes.

C. Object Properties

Semantic relationships are established through object properties such as hasAttendee (linking events like ABsWedding to attendees such as Aftab_Ahmed), hasVendor (associating EliteCatering with events), and hasVenue (binding ABsWedding to GrandHallVenue). Specialized properties like performsAt (connecting performers to events) and speakAt (linking speakers to conferences) enable granular role-event associations. Temporal constraints are enforced via hasSchedule, which ties events to schedules like MorningTimeABsWedding.

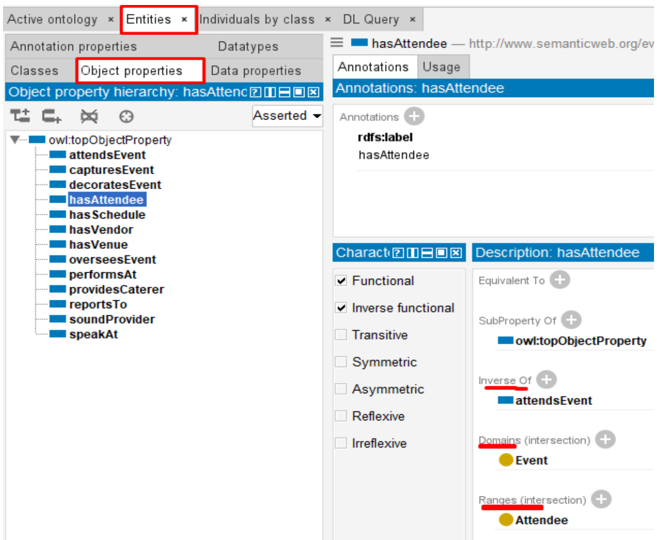

Fig. 8. Event Management System Ontology Object Properties

TABLE I. OBJECT PROPERTY OF SEMANTIC REASONING IN EVENT MANAGEMENT SYSTEM ONTOLOGY

Object Property	Domain	Range	Inverse Property
attendsEvent	Attendee	Event	hasAttendee
capturesEvent	Photographer	Event	-
decoratesEvent	Decorator	Event	-
hasAttendee	Event	Attendee	attendsEvent
hasSchedule	Event	Schedule	-
hasVendor	Event	Vendor	-
hasVenue	Event	Venue	-
OverseesEvent	OrganizationalRole	Event	-
PerformsAt	Performer	Event	-
providesCaterer	Caterer	Event	-
soundProvider	SoundSystemProvider	Event	-
speakAt	Speaker	Event	-

D. Data Properties

Capture essential attributes: eventDate (of type xsd:dateTime) records dates, attendeeName and attendeeEmail store participant details, and hasSalary (of type xsd:decimal) manages salaries for roles like FinanceManager. Venue-specific attributes, such as venueCapacity (integer), are also defined.

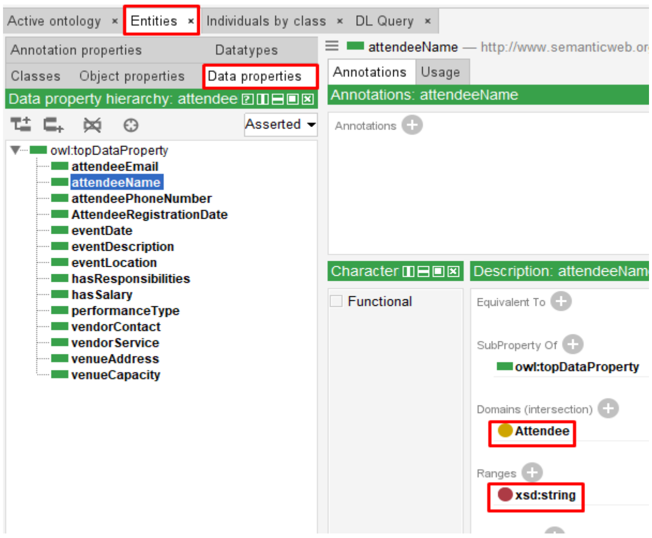

Fig. 9. Event Management System Ontology Data Properties

TABLE II. DATA PROPERTY FOR SEMANTIC REASONING IN EVENT MANAGEMENT SYSTEM

Data Properties	Domains	Range
attendeeEmail	Attendee	xsd:string
attendeeName	Attendee	Xsd:string
attendeePhoneNumber	Attendee	xsd:string

eventDate	Event	xsd:dateTime
eventDescription	Event	xsd:string
eventLocation	Event	xsd:string
hasResponsibilities	OrganizationalRole	xsd:string
hasSalary	OrganizationalRole	Xsd:decimal
performanceType	Performer	xsd:string
vendorContact	Vendor	xsd:string
vendorService	Vendor	xsd:string
venueAddress	Venue	xsd:string
venueCapacity	Venue	xsd:integer

E. *Individuals*

The ontology is validated through instantiated individuals representing real-world scenarios. For example, ABsWedding (a Wedding event) uses GrandHallVenue (an Indoor venue) via hasVenue and is linked to vendors like EliteCatering (via providesCaterer) and PhotographyPro (via capturesEvent). Attendees such as Aftab_Ahmed (VIP_Guest) and Sheraz (Performer) demonstrate role-specific participation. Logical consistency is ensured through disjoint axioms (e.g., Event and Venue cannot overlap) and inverse properties (e.g., attendsEvent and hasAttendee).

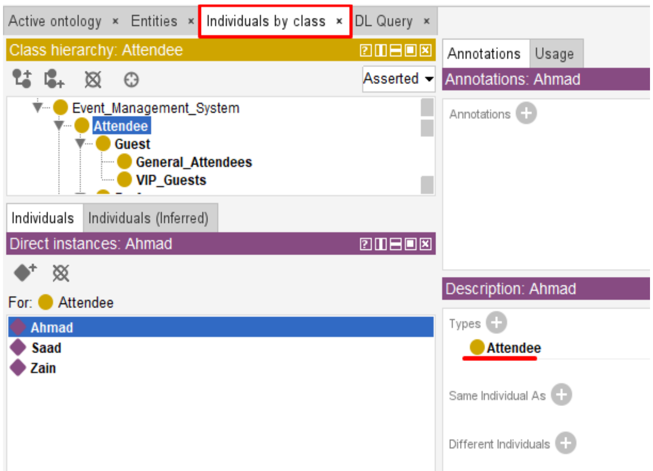

Fig. 10. Event Management System Ontology Individuals

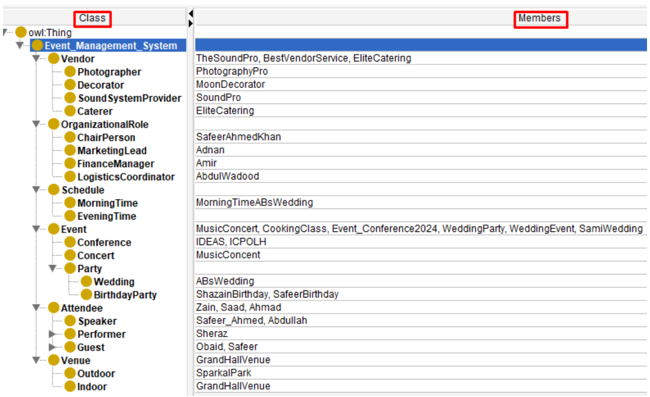

Fig. 11. Event Management System Ontology Individuals matrix

V. EVALUATION: INFERRED KNOWLEDGE AND QUERYING THE ONTOLOGY

A. *INFERRED KNOWLEDGE*

The ontology's hierarchical structure enables automated reasoning to derive implicit knowledge, such as classifying BirthdayParty as an Event through the inheritance chain where BirthdayParty is a subclass of Party, which is a subclass of Event. Roles like DancePerformer are recognized as Attendees via Performer being a subclass of Attendee. Vendor subclasses, including Caterer and Decorator, are systematically categorized under Vendor. Logical constraints, such as disjoint axioms (e.g., Event and Venue are distinct classes to prevent misclassifications like GrandHallVenue as an Event) and property rules (e.g., mandatory hasSpeaker for Conference), enforce domain-specific consistency. Key outcomes include automated classification of events, roles, and vendors; validation of data types (e.g., decimal salaries, integer venue capacities); and conflict detection (e.g., double-booked venues).

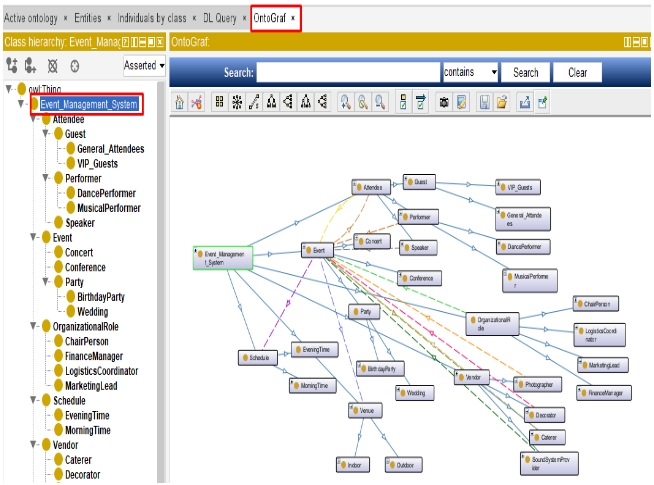

Fig. 12. Event Management System Ontology Hierarchy Graph

B. *QUERYING THE ONTOLOGY*

The DL query retrieves individuals who are either Speakers at the IDEAS Conference or Performers at ABsWedding, using the logical OR (∨) operation. As shown in Fig. 6, the query results include Safeer_Ahmed and Sheraz.

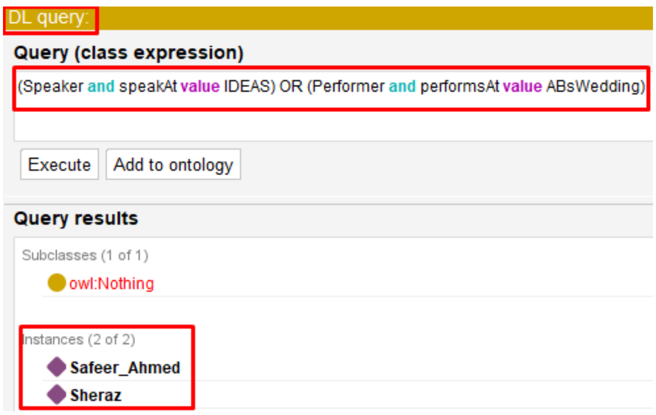

Fig. 13. Event Management System Ontology DL query (1)

The DL query retrieves individuals who are Performers at both ICPOLH Conference and ABsWedding using the AND (∧) operator. As shown in Fig. 7, Sheraz is the only instance meeting this condition.

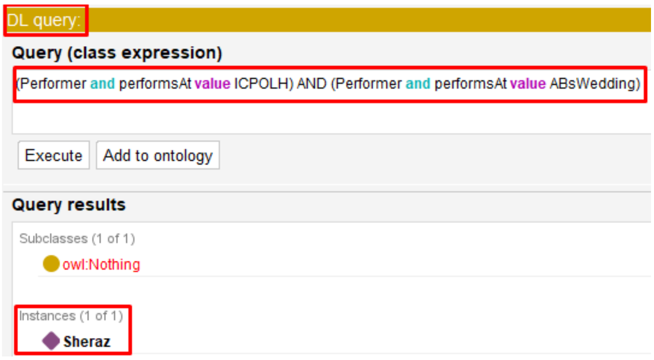

Fig. 14. Event Management System Ontology DL query (2)

The DL query retrieves Photographer instances linked to the event ABsWedding via the capturesEvent property. It identifies PhotographyPro as the vendor fulfilling this role.

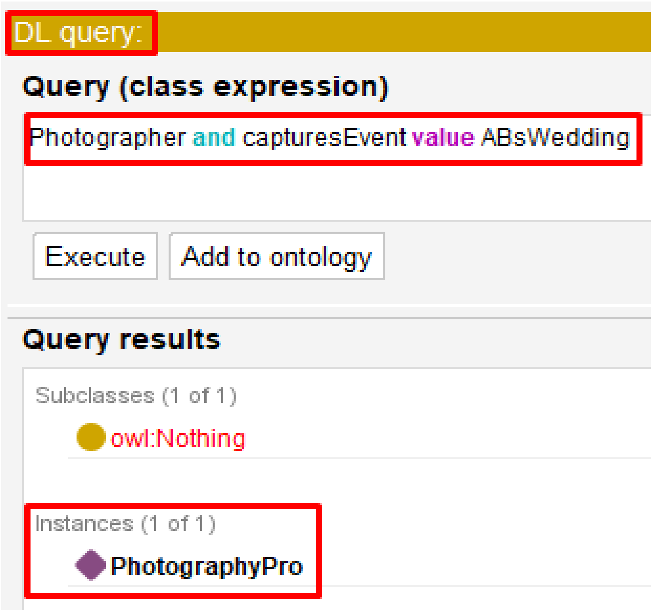

Fig. 15. Event Management System Ontology DL query (3)

The DL query retrieves Guest instances linked to the event ABsWedding via the attendsEvent property. It returns seven attendees, including Aamir, Safeer, and Zeeshan, fulfilling this role.

Fig. 16. Event Management System Ontology DL query (4)

The DL query retrieves Caterer instances linked to both ABsWedding and SafeerBirthday via the providesCaterer property. It identifies EliteCatering as the vendor fulfilling both events.

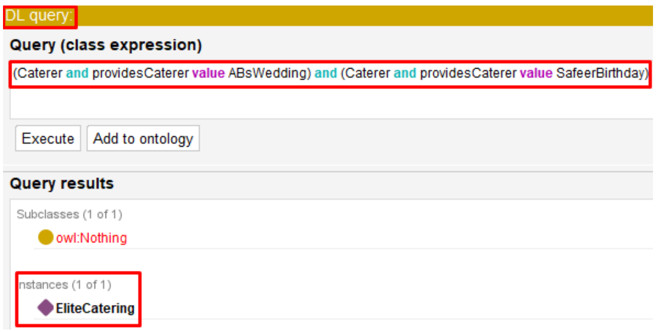

Fig. 17. Event Management System Ontology DL query (5)

This DL query retrieves all events that have both a **Photographer** and a **Decorator** assigned as vendors. It ensures that the hasVendor property links the event to both service provider types.

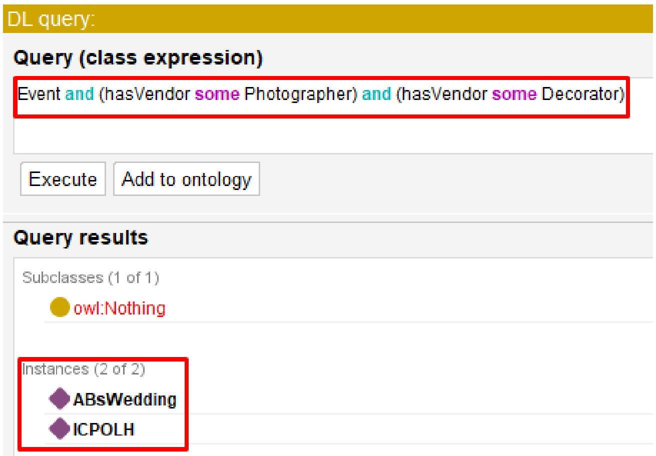

Fig. 18. Event Management System Ontology DL query (6)

This DL query retrieves all **Event** instances that meet three specific conditions, they must be held at an **Outdoor** venue, include at least one **VIP_Guests,** attendee and involve a **SoundSystemProvider** as a vendor.

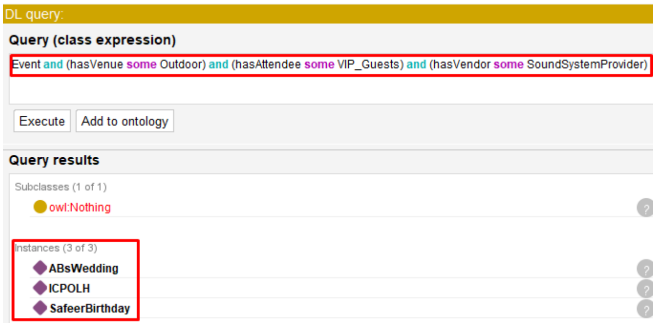

Fig. 19. Event Management System Ontology DL query (7)

VI- CONCLUSIONS

The Semantic Reasoning in Event Management Ontology using OWL excels in the systematic arrangement of components, roles, and interactions related to events. It maintains uniformity through a definite hierarchy composed of core classes such as Event, Attendee, and Vendor which form the important constituent groups. Relationships within the subclass Marriage combines with Party and Performer merges with Attendee clarify specialized roles. Connections such as attendsEvent and hasAttendee automagically bind participants to events. Other properties like hasVendor and hasVenue aid in resource scope such as ABsWedding in relation to vendors like EliteCatering and the GrandHallVenue. Conflicts are avoided through logical rules like maintaining a separation between Event and Venue. Supplementary structure is offered by data properties eventDate and venueCapacity while FinanceManager in organizational role provides clarity through attributes hasSalary and hasResponsibilities. Control is partially provided through time subdivision under Schedule into MorningTime and EveningTime.

However, the ontology has limitations. To enhance scalability and adaptability, future iterations could integrate startTime and endTime for precise scheduling, add Ticket and paymentMethod classes for financial workflows, and categorize vendors into tiers like PremiumCaterer or StandardCaterer. Expanding properties such as reportsTo would streamline team collaboration, while enabling venues to host multiple events and introducing wheelchairAccessible or lastUpdatedTime properties would improve inclusivity and real-time tracking. These enhancements would strengthen the ontology's utility for diverse scenarios, from conferences to festivals, ensuring robust support for modern event management needs.

REFERENCES

[1] Strategic planning kit for dummies. https://books.google.com/books?hl=en&lr=&id=3P-t78Bosq4C&oi=fnd&pg=PA1&dq=Strategic+planning+kit+for+dummies&ots=3L11od7qiV&sig=ii68gr5UdWPVYp2VhD5ixBLJBfc

[2] S. Madanska, "An ontology for architectural heritage: historical figures and organizations", Digital Presentation and Preservation of Cultural and Scientific Heritage, vol. 12, p. 121-130, 2022. https://doi.org/10.55630/dipp.2022.12.9

[3] P. Sinha, S. Gajbe, S. Debnath, S. Sahoo, K. Chakraborty, & S. Mahato, "A review of data mining ontologies", Data Technologies and Applications, vol. 56, no. 2, p. 172-204, 2021. https://doi.org/10.1108/dta-04-2021-0106

[4] A. Alahmar and R. Benlamri, "Snomed ct-based standardized e-clinical pathways for enabling big data analytics in healthcare", Ieee Access, p. 1-1, 2020. https://doi.org/10.1109/access.2020.2994286

[5] E. Alkhammash, "Graphical transformation of owl ontologies to event-b formal models", Computers Materials & Continua, vol. 70, no. 2, p. 3733-3750, 2022. https://doi.org/10.32604/cmc.2022.015987

[6] Q. Liu, J. Wang, Y. Zhu, & Y. He, "Ontology-based systematic representation and analysis of traditional chinese drugs against rheumatism", BMC Systems Biology, vol. 11, no. S7, 2017. https://doi.org/10.1186/s12918-017-0510-5

[7] R. Chen, H. Jiang, C. Huang, & C. Bau, "Clinical decision support system for diabetes based on ontology reasoning and topsis analysis", Journal of Healthcare Engineering, vol. 2017, p. 1-14, 2017. https://doi.org/10.1155/2017/4307508

[8] R. Ortega, E. Barreiro, & M. Fortuna, "Ontology for the avida digital evolution platform", Scientific Data, vol. 10, no. 1, 2023. https://doi.org/10.1038/s41597-023-02514-3

[9] V. Lytvyn, V. Vysotska, D. Dosyn, O. Lozynska, & O. Oborska, "Methods of building intelligent decision support systems based on adaptive ontology",, p. 145-150, 2018. https://doi.org/10.1109/dsmp.2018.8478500

[10] M. Sanati and A. Asadi, "Satisfiability checking of clinical practice guidelines using an analyzer", Modern Care Journal, vol. 17, no. 1, 2020. https://doi.org/10.5812/modernc.98204

[11] H. Massari, N. Gherabi, S. Mhammedi, H. Ghandi, M. Bahaj, & M. Naqvi, "The impact of ontology on the prediction of cardiovascular disease compared to machine learning algorithms", International Journal of Online and Biomedical Engineering (Ijoe), vol. 18, no. 11, p. 143-157, 2022. https://doi.org/10.3991/ijoe.v18i11.32647

[12] e. Naz, "Fully automatic owl generator from rdb schema", International Journal of Advanced and Applied Sciences, vol. 5, no. 4, p. 79-86, 2018. https://doi.org/10.21833/ijaas.2018.04.010

[13] H. Elazhary, "Cal: a controlled arabic language for authoring ontologies", Arabian Journal for Science and Engineering, vol. 41, no. 8, p. 2911-2926, 2016. https://doi.org/10.1007/s13369-015-2016-z

[14] **N. F. Noy and D. L. McGuinness**, "Ontology Development 101: A Guide to Creating Your First Ontology," *Stanford University Technical Report*, 2001. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[15] Evento Pakistan, "Evento – Event management & Planners," https://evento.com.pk/

[16] A2z Events Solutions - Lahore, Pakistan https://a2zeventssolutions.com/

VII. APPENDIX

Event Management System Ontology by