



PROPUESTA DE UN MODELO ONTOLÓGICO EN SISTEMAS FEDERADOS EN BASE A TAXONOMÍAS JC3IEDM

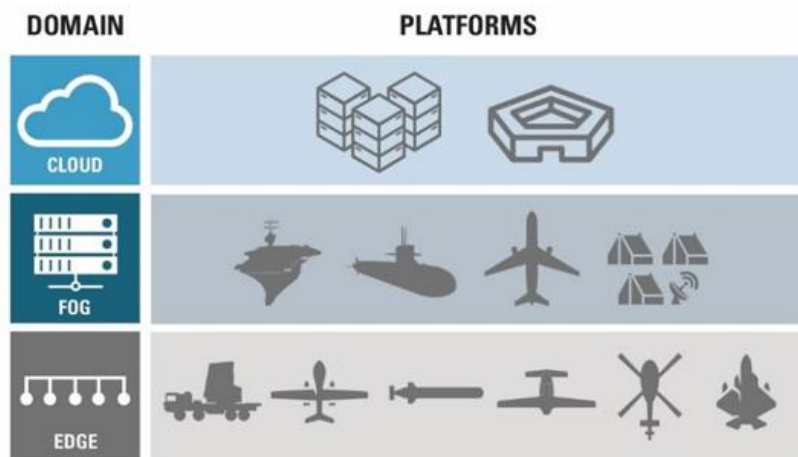
CARLOTA MARÍA MUÑOZ DE LUNA EUSEBIO
CÁTEDRA ISDEFE-UPM

I. CONTEXTO

En los últimos años, con el crecimiento de la Inteligencia Artificial (IA) y el Machine Learning (ML), se han planteado problemas sobre cómo se utiliza la información durante su entrenamiento y ha llevado a la necesidad de modificar las regulaciones sobre la privacidad. Este cambio en las regulaciones está relacionado con el crecimiento exponencial de la IA y el ML, y con la forma en que estos tratan la información que utilizan para su entrenamiento.

De manera similar en el ámbito de la defensa se busca mantener la privacidad de los datos especialmente cuando se requiere la colaboración entre múltiples entidades. En el ámbito de la defensa, donde la información es sensible y clasificada, las arquitecturas centralizadas de ML pueden no ser adecuadas debido a la necesidad de preservar la privacidad y mantener la independencia de los datos. La aparición de un nuevo paradigma llamado **Combat Cloud** o Nube de Combate [1] está transformando la manera en que se conciben las operaciones militares. La nube de combate emerge como respuesta a la necesidad de operar en red para alcanzar superioridad en entornos multidominio (tierra, mar, aire, espacio y ciberespacio) [2] mediante un conocimiento completo de la situación, a través de la captura, compartición, fusión y procesamiento instantáneo de grandes cantidades de datos de todos los activos conectados, para apoyar decisiones precisas y ágiles, consiguiendo una superioridad de información.

La infraestructura de la Nube de Combate se divide en tres dominios. El dominio “Cloud” maneja grandes cantidades de datos en sistemas con alta capacidad de computación, cuya ubicación no es crucial para la operación. Posteriormente, “Fog” actúa como intermediario entre el “Cloud” y “Edge”, manejando menos datos con una capacidad limitada de computación y en ubicaciones cercanas al entorno de operación. Finalmente, el dominio “Edge” incluye sistemas con efectores y/o sensores que proporcionan datos de nivel inferior consumidos por los dominios superiores para generar información más avanzada.



A pesar de la efectividad probada de los modelos de ML, su aplicación práctica conlleva una serie de desafíos. A menudo es necesario un nodo o dispositivo centralizado que procese grandes volúmenes de datos para el entrenamiento de los algoritmos. Esta arquitectura puede presentar dificultades en la fase de recolección de datos para el entrenamiento debido a que, en ocasiones, las muestras obtenidas pueden no ser suficientes para entrenar un modelo eficaz o, incluso, los datos necesarios podrían no estar disponibles para su utilización.

En el contexto de los modelos de detección de objetos basados en ML, estos problemas se manifiestan en determinados momentos. Por ejemplo, durante la recolección de imágenes provenientes de vehículos aéreos no tripulados (UAV) en diversas ubicaciones, la acumulación de carga de tráfico en el nodo central para recibir las diferentes imágenes puede desembocar en problemas como baja latencia de red o incluso pérdida de paquetes.

En dichas instituciones enmarcadas en el ámbito de la defensa, surgen obstáculos relacionados con la privacidad o la incompatibilidad entre los conjuntos de datos generados. Las instituciones podrían manejar información sensible que no pueden compartir a un nodo central debido a restricciones legales o para mantener ventajas competitivas. Además, cada organización puede extraer características diferentes de sus datos, lo que dificulta su combinación y análisis conjunto.

Ante esta problemática, se plantea explorar cómo sería la implementación de Aprendizaje Federado o Federated Learning (FL) para entrenar un modelo de detección de objetos en imágenes. Este método consiste en el entrenamiento de modelos de ML de forma colaborativa, donde a diferencia de los enfoques tradicionales, que realizan el proceso de entrenamiento en un lugar centralizado, se envía el modelo a los clientes donde se encuentran los datos requeridos.

A pesar de las ventajas del FL en términos de privacidad, surge la necesidad de estructurar y formalizar los datos que se van a utilizar. En este contexto, **JC3IEDM** (Joint Consultation, Command and Control Information Exchange Data Model) es un estándar ampliamente adoptado en operaciones militares para asegurar una representación coherente de los datos entre entidades. Sin embargo, JC3IEDM tiene limitaciones en cuanto a la flexibilidad y semántica, dificultando su adaptación a sistemas autónomos y distribuidos como los requeridos en FL. Para abordar estos desafíos, proponemos el uso de **OWL (Web Ontology Language)** como una mejora semántica que permite la representación y manejo de conocimiento de una manera más rica y adaptable.

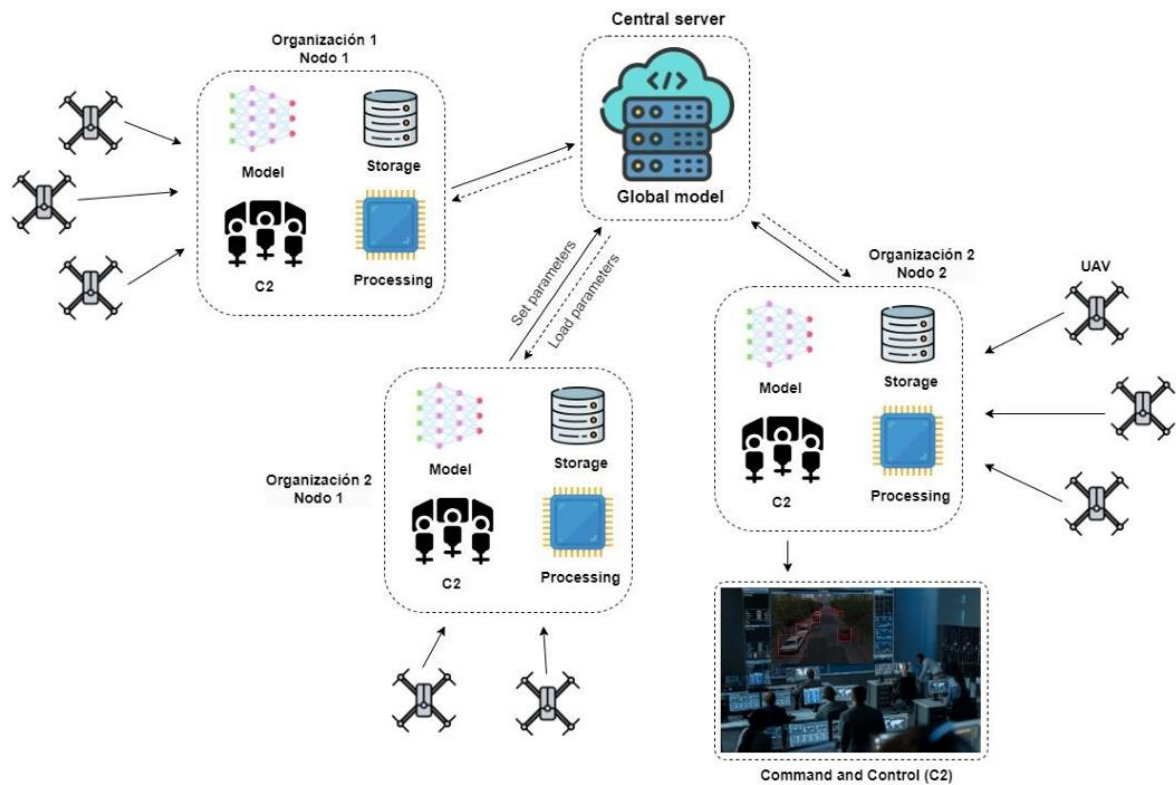
II. TFT

El auge de las técnicas de DL y ML ha impulsado a que el entrenamiento de estos modelos se lleve a cabo en lugares centralizados con acceso a grandes volúmenes de datos, esenciales para maximizar el rendimiento de estos modelos.

Durante el entrenamiento de modelos de Machine-Learning el volumen de datos generados es inmenso y está distribuido en millones de dispositivos y sistemas independientes. Esto requiere la centralización de los datos en un servidor, lo que implica transferir grandes cantidades de información a una ubicación única. En el ámbito de defensa esta metodología plantea varios problemas como la privacidad y seguridad de los datos.

Se propone diseñar e implementar un sistema de detección de objetos mediante UAV basado en modelos de aprendizaje federado.

El modelo aprenderá de las imágenes recogidas por cada nodo participante, permitiendo un entrenamiento colaborativo para desarrollar un modelo global de detección de objetos. Este enfoque de entrenamiento permite la colaboración entre organizaciones independientes, preservando la privacidad de sus datos sensibles y evitando regulaciones específicas de cada país.



En la figura anterior podemos observar varios componentes:

- Tres nodos clientes que pueden pertenecer a la misma o a diferentes organizaciones con diversas capacidades de almacenamiento y procesamiento, y elementos como el modelo obtenido de FL. Estos nodos clientes son responsables de entrenar sus propios modelos locales y compartir sus parámetros con el servidor central.
- Un puesto de Command and Control(C2) donde se coordina toda la información recibida del campo de operaciones. En este caso, se reciben videos en tiempo real de los UAVs desplegados, y tras procesar y extraer los fotogramas, se aplica el modelo de detección de objetos global entrenado para visualizar las detecciones y extraer información valiosa.
- Un servidor central ubicado en la nube que actúa como coordinador del entrenamiento del modelo de FL. Se encarga de recoger los parámetros del modelo global a los nodos en cada una de las rondas de entrenamiento, definiendo las rondas, numero de épocas, número de clientes y la estrategia de agregación utilizada

III. PROPUESTA

Para lograr que entidades diversas colaboren en diferentes entornos, es necesario un modelo de datos coherente que facilite la comprensión y el intercambio de información entre ellas.

JC3IEDM (Joint Consultation, Command and Control Information Exchange Data Model) es un modelo de datos relacional diseñado para facilitar la interoperabilidad y el intercambio de

información entre sistemas militares de distintos países y organizaciones. Este modelo puede analizarse desde tres perspectivas distintas: conceptual, lógica y física.

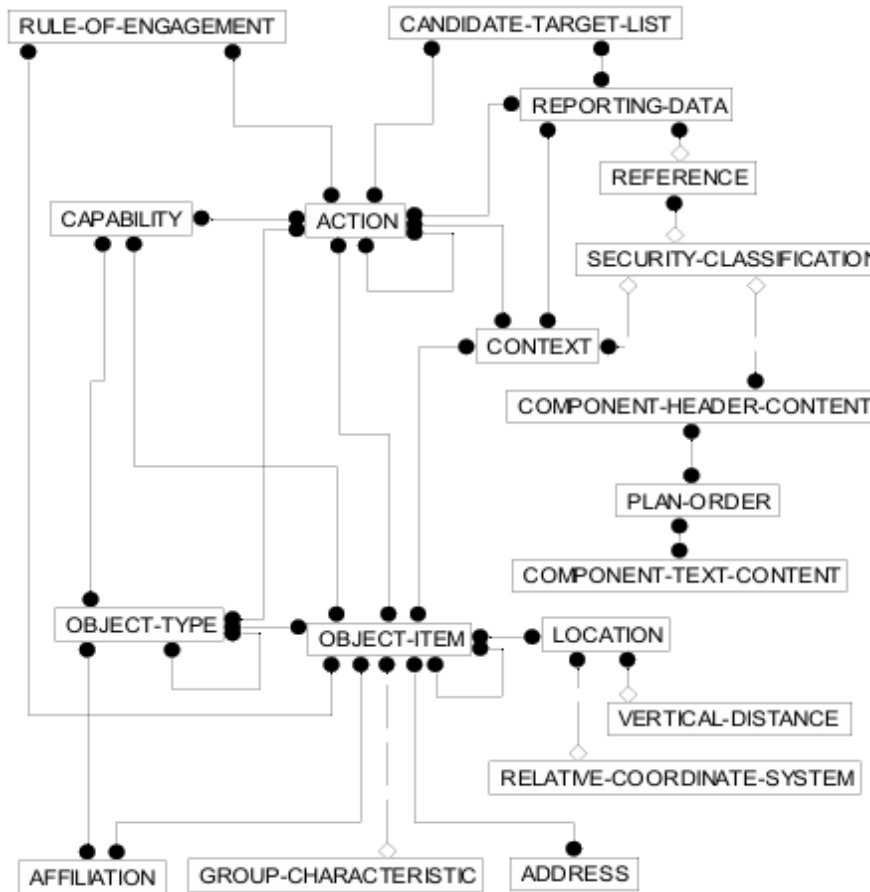
1. **Modelo Conceptual:** Es una vista abstracta que define los elementos de datos más importantes (por ejemplo, acciones, personal, instalaciones) y su propósito es proporcionar una comprensión general sobre el alcance y el contenido del modelo. Este nivel es útil para obtener una visión amplia de los tipos de datos que se manejan en el sistema.
2. **Modelo Lógico:** Añade detalles específicos necesarios para comprender las conexiones lógicas entre los elementos del modelo. Su objetivo es crear una representación que sea comprensible y accesible para las personas, es decir, que permita captar las relaciones y la estructura lógica sin preocuparse por la implementación física. Esta es la perspectiva más relevante para el estudio del modelo, ya que permite comprender la semántica lógica sin necesidad de adentrarse en aspectos técnicos de la implementación.
3. **Modelo Físico:** Este nivel está enfocado en los detalles necesarios para implementar el modelo de datos como un esquema de base de datos. A partir del modelo lógico, el modelo físico incorpora claves y datos redundantes para optimizar el rendimiento.

El modelo puede verse de manera abstracta como un metamodelo cuyo esqueleto principal se basa en los siguientes temas:

- **Objetos** de interés y sus propiedades. Estas propiedades pueden ser instalaciones, elementos, materiales, organizaciones y personas identificadas unívocamente como elementos o utilizadas según sus características de clase o tipo
- **Situaciones** pasadas, presentes o futuras sobre objetos. Esto incluye relaciones *type-to-type*, *item-to-type*, capacidades que tienen algunos objetos o tipos de objetos, afiliaciones entre objetos o tipo de objetos, estado, localización, direccionamiento, así como relaciones entre dichos objetos.
- **Actividades** pasadas, presentes o futuras que involucran objetos. La especificación básica de la actividad describe cómo se usan los objetos como recursos, objetivos o efectos de una actividad. Además, el modelo incluye extensiones para detallar la actividad de varias maneras, como, por ejemplo, las reglas de enfrentamiento y la creación de listas de posibles objetivos.
- Mecanismos de agrupación de la **información en paquetes**. El modelo tiene una estructura llamada REPORTING-DATA (Datos de Informe) que se relaciona con la mayoría de las instancias de datos dinámicos. Estas especificaciones permiten que conjuntos de registros individuales se traten como un paquete, al que se le denomina "contexto". Esta estructura de contexto tiene múltiples usos y puede estar vinculada tanto a elementos como a actividades. Además, permite adjuntar evaluaciones a un contexto. Finalmente, hay una especificación que permite añadir información descriptiva de varios tipos a grupos de personas.

La información se estructura a partir de entidades y relaciones. La **entidad** es cualquier persona, lugar, objeto, evento o concepto distinguible sobre el cual se va a almacenar información. Cada entidad se define a través de sus propiedades o características, conocidas como **atributos**, que especifican los datos que deben registrarse para cada entidad. Estas entidades pueden ser tanto independientes como dependientes. Las entidades independientes forman la base del modelo, y las dependientes se construyen en función de ellas.

Las entidades se relacionan entre ellas. Una **relación** en el modelo describe cómo dos entidades están conectadas entre sí. Puede ser uno-a-muchos (representada por una línea discontinua) o muchos-a-muchos (representada por una línea continua con puntos en ambos extremos). Una instancia de una entidad puede estar relacionada con múltiples instancias de otra y viceversa. Para garantizar la integridad de los datos, en los modelos de datos completamente desarrollados, las relaciones muchos-a-muchos se resuelven en relaciones uno-a-muchos mediante la creación de nuevas estructuras intermedias.



En la imagen anterior podemos ver como las entidades independientes se relacionan entre ellas. A continuación, se proporciona un breve resumen de cada una:

- Las **ACTION**: Representa una actividad u ocurrencia que puede usar recursos y estar dirigida a un objetivo. Ejemplos incluyen órdenes y planes de operaciones, movimientos, misiones de apoyo aéreo, y eventos como ataques enemigos. Proporciona información sobre cómo, qué y cuándo algo debe realizarse o se ha realizado.
- **ADDRESS**: Información detallada, física o electrónica, que permite acceder a un destino. Incluye direcciones postales y electrónicas.
- **AFFILIATION**: Especifica la asociación de un objeto con un país, grupo funcional, religión, etc. Permite asignar afiliaciones a tipos de objetos o elementos específicos.
- **CANDIDATE TARGET-LIST**: Lista de objetos o tipos en el espacio de batalla que podrían ser destruidos o explotados. Es propuesta por una autoridad para considerarse en actividades de planificación.

- **CAPABILITY:** Describe la capacidad potencial de realizar una función, misión u objetivo. Indica capacidades esperadas y reales para tipos de objetos e ítems específicos.
- **COMPONENT-HEADER CONTENT:** Información introductoria que identifica elementos de un plan u orden. Se utiliza junto con especificaciones de planes y órdenes.
- **COMPONENT-TEXT CONTENT:** Declaración textual que contiene información esencial. También se utiliza junto con especificaciones de planes y órdenes.
- **CONTEXT:** Colección de información que define las circunstancias o entorno de una situación, permitiendo agrupar datos de manera coherente.
- **RELATIVE COORDINATE SYSTEM:** Sistema de coordenadas rectangular que define un marco de referencia con un origen y ejes (x, y, z), utilizado para especificar geometría relativa.
- **GROUP CHARACTERISTIC:** Conjunto de características que permite identificar una colección de objetos, como grupo de edad, género, idioma, etc. Facilita el conteo de personas según características seleccionadas.
- **LOCATION:** Especifica posición y geometría en relación con un marco de referencia, como puntos, círculos, áreas, etc., proporcionando detalles geográficos y dimensiones.
- **OBJECT-ITEM:** Objeto identificado individualmente con relevancia militar o civil, como una persona, unidad, o característica geográfica específica. Identifica entidades individuales.
- **OBJECT-TYPE:** Clase de objetos con relevancia militar o civil, como un tipo de material o unidad. Identifica clases de entidades.
- **PLAN-ORDER:** Esquema predefinido para alcanzar un objetivo operativo. Es la entidad principal para identificar un plan u orden.
- **REFERENCE:** Describe la fuente de información con importancia militar o civil, apoyando datos de reporte.
- **REPORTING-DATA:** Información externa que respalda los datos de reporte, especificando la fuente, calidad y momento de los datos.
- **RULE-OF ENGAGEMENT:** Directrices obligatorias sobre cómo debe ejecutarse una actividad específica, útil para funciones de reporte.
- **SECURITY CLASSIFICATION:** Clasificación de seguridad aplicada a recursos informativos en un contexto de información clasificada.
- **VERTICAL DISTANCE:** Define la altitud o altura de un punto respecto a un datum, ayudando a especificar la elevación en ubicaciones.

Sin embargo, debido a la complejidad del modelo, se proponen algunos puntos a mejorar [4]

- Alcance. El alcance actual, que abarca todo el C3 conjunto de la OTAN, es demasiado amplio, lo que lo vuelve complejo y difícil de manejar. Se recomienda dividirlo en modelos más pequeños y específicos, como uno para C2 terrestre, otro para estandarización ("paraguas semántico") y otros para C2 aéreo y marítimo.
- Complejidad: El JC3IEDM es muy grande y complejo, con 195 entidades, 785 atributos y 277 relaciones. La documentación que lo acompaña supera las 1000 páginas, lo que dificulta su comprensión e implementación.
- Flexibilidad: Si bien el modelo presenta cierta flexibilidad, la gran cantidad de partes interesadas y los intereses no relacionados con el modelado (políticos, financieros, comerciales) dificultan la incorporación de cambios y actualizaciones
- Datos Estáticos versus Dinámicos: Los datos estáticos (ej: características de un tipo de avión) no deberían incluirse en el IEDM, ya que rara vez cambian. Se recomienda

gestionarlos por separado y referenciarlos a través de dominios para que, en lugar de eliminarlos en su totalidad, puedan proporcionar la misma información sin complicar en exceso el esquema.

El JC3IEDM incluye estructuras estáticas, como OBJECT-TYPE, ESTABLISHMENT y CAPABILITY, que deberían eliminarse o simplificarse.

- Ítem versus Tipo. Los OBJECT-ITEM hacen referencia a instancias mientras que OBJECT-TYPE a categorías abstractas. Se recomienda utilizar los tipos con moderación, ya que suelen contener datos estáticos. En el caso del JC3IEDM, la estructura de OBJECT-TYPE parece excesivamente grande y estática, lo que la hace poco útil para el intercambio de información.
- Estructuras genéricas vs. explícitas: Se observa un desequilibrio en el uso de estructuras genéricas y explícitas. Se plantea el dilema entre modelar la información de forma explícita, representando los objetos del mundo real de forma literal en el modelo de datos, o de manera genérica, utilizando estructuras más abstractas y flexibles.
 - **Ventajas de lo Explícito:** Facilidad de comprensión, menor necesidad de restricciones explícitas, implementación más directa.
 - **Ventajas de lo Genérico:** Estructuras más compactas, mayor flexibilidad y estabilidad ante cambios, reutilización en diferentes contextos.

La recomendación es utilizar estructuras **explícitas** para conceptos comunes y fundamentales (ej: AIRCRAFT, AIRBASE, SQUADRON), y estructuras **genéricas** para aspectos menos comunes o más específicos (ej: pilot modelado como PERSON, squadron-crewchief como una asociación entre UNIT y PERSON).

El JC3IEDM, en general, tiende a ser genérico, lo que facilita la incorporación de conceptos aéreos y marítimos en un modelo originalmente terrestre. Sin embargo, esto puede dificultar su aceptación por parte de los usuarios de la Fuerza Aérea y la Armada, ya que no reconocerían fácilmente sus datos específicos en el modelo.

- Normalización: La normalización evita la duplicación de datos y las inconsistencias, pero puede resultar en modelos complejos y difíciles de manejar. JC3IEDM presenta ejemplos de estructuras que podrían desnormalizarse, como la sección de LOCATION (con 22 entidades), los subtipos de OBJECT-ITEM-STATUS y AFFILIATION, y el árbol de subtipos de OBJECT-TYPE.
- Metadatos: Los metadatos son datos sobre los datos, como información sobre su creación, distribución, modificación, seguridad y calidad. El artículo desaconseja incluir metadatos en el esquema lógico del IEDM, ya que lo hacen innecesariamente complejo. Se recomienda gestionarlos por separado, ya sea añadiéndolos al esquema físico o definiéndolos en un modelo de metadatos independiente.

El JC3IEDM incorpora algunos metadatos en el esquema físico (ej: owner-id), y utiliza un modelo de replicación para gestionar otros aspectos, como los nodos y contratos. Sin embargo, la entidad REPORTING-DATA introduce metadatos innecesarios en el modelo
- Entidades independientes: Las entidades independientes no dependen de otras entidades para su existencia. Se recomienda limitar su uso a los objetos del mundo real más fundamentales, reconocibles y concretos. Por ejemplo

Por ejemplo, las piezas de MATERIEL (independientes) existen por sí solas, pero los datos de MATERIEL-STATUS (dependientes) sólo pueden existir cuando están relacionados con una instancia específica de MATERIEL. El JC3IEDM incluye algunas entidades independientes que podrían redefinirse como propiedades de otras entidades, como AFFILIATION, CANDIDATE-TARGET-LIST, LOCATION y, posiblemente, CAPABILITY.

- Información no estructurada: La capacidad del JC3IEDM para integrar información no estructurada es limitada. Se recomienda incorporar atributos que referencien documentos, imágenes y otros tipos de datos no estructurados

Sin embargo, aunque JC3IEDM es robusto en términos de representación estructurada, tiene limitaciones para adaptarse a los sistemas autónomos y distribuidos a los que requiere FL.

Es aquí donde el **Web Ontology Language (OWL)** entra en juego. Al combinar JC3IEDM con OWL, podemos enriquecer el modelo de datos existente, permitiendo una representación más flexible y semánticamente rica. OWL proporciona la base para definir relaciones, atributos y contextos específicos de manera que todas las entidades puedan interpretar la información de manera precisa, aun cuando los datos no se centralicen.

Para ello vamos a hacer uso de la vista lógica de JC3IEDM ya que este incluye la estructura conceptual de las entidades y relaciones sin entrar en detalles de implementación física.

Todos los elementos a utilizar los vamos a agrupar en:

- Entity_Groups: Incluye entidades específicas
- Attribute_Groups: Incluye los atributos específicos de las entidades
- Relationship_Groups: Contiene las relaciones que pueden darse entre entidades
- Domain_Groups and Validation_Rule_Groups: Contiene los valores que se pueden poner en los dominios y rangos de los atributos asegurando que los datos sigan un conjunto de valores específicos y sean válidos dentro de las restricciones definidas.

Muchas de las entidades existen como subclases de al menos otra entidad. Sabemos que una Entidad es una subclase de otra si la entidad padre tiene como atributo un **Valid_Value** en el que se incluye la entidad hijo y además esta entidad hijo tiene un atributo *category-code* de la entidad padre.

Estos elementos se van a transformar en:

- Entity_Group → owl:Classes
- Attribute_Groups → owl:ObjectProperty o owl:DatatypeProperty
- Relationship_Groups → owl:ObjectProperties
- Entidad hijo → owl: subclassOf

A. CLASSES

Hay que tener en cuenta que no todas las clases de objetos son definidas como entidades. Solo se consideran aquellas que tienen atributos o relaciones.

Las entidades que tienen subclases de las mencionadas anteriormente tienen un atributo caracterizado por estar escrito de la forma *nombrentidad-category-code*.

Cada clase necesita un *rdf:ID* único. Normalmente se pone el nombre de la entidad, pero a veces el mismo *category-code* (por ejemplo, "Coche") puede usarse en más de una entidad (es decir tendría el mismo *rdf:ID*). En estos casos por ejemplo si "Coche" ya existe en otra entidad y se vuelve a usar en una segunda entidad, entonces añadiremos el nombre de la clase padre (en este caso "Vehículo") delante de "Coche" para hacer el *rdf:ID* único.

B. PROPERTY

Se establecen criterios para determinar si un atributo se convierte en un *owl:ObjectProperty* o un *owl:DatatypeProperty*. Vamos a excluir aquellos que posean algunos de los siguientes caracteres en el nombre del atributo:

- *category-code*: Ya que como hemos visto los *category-code* se van a traducir a Entidades → *owl:Classes*
- *id*: Ya que ya se está usando un *rdf*
- *-index* y *-ent_cat_code*: se excluyen porque forman parte de la parte física del modelo
- *-update_seqnr* se suele usar para manejar replicaciones por lo que también lo descartaremos.
- *-uodate_seqnr* no se usa porque a veces se puede confundir con que hay una errata

Algunos atributos representan relaciones entre "cosas" o categorías. Si un atributo usa un **category-code**, se convierte en un **ObjectProperty**. Esto significa que conecta una cosa (como un tipo de vehículo) con otra cosa (como categorías de vehículos: coche, camión, etc.). Otros atributos solo almacenan datos simples, como números o texto. Estos se convierten en **DatatypeProperties**.

C. OBJECTPROPERTY

Cada relación conecta dos clases. Esta se identifica por un identificador único. El modelo JC3IEDM a menudo reutiliza los mismos nombres de relaciones para distintas conexiones (como "has" o "is-object-of"). Si simplemente se usara el nombre, podrían surgir **conflictos** de nombres, donde varias relaciones tendrían el mismo nombre y OWL no podría distinguirlos. Hay dos opciones para solucionar esto:

- **Crear clases de unión** que combinen múltiples clases. Esto permitiría usar los mismos nombres para relaciones diferentes, pero causaría confusión porque podría conectar clases que no deberían estar relacionadas.
- **Crear un esquema de nombres único** que haga los nombres más largos y detallados para cada relación.

Tendremos en cuenta la segunda opción porque mantiene el significado de cada relación y evita confusiones. Esto también es una desventaja porque da como resultado nombres muy largos por ejemplo (OPERATIONAL-INFORMATION-GROUP-ORGANISATIONASSOCIATION-has-OPERATIONAL-INFORMATION-GROUP-ORGANISATION-ASSOCIATIONSTATUS) para solucionar esto vamos a tomar solo sus iniciales (OIGOA-has-OIGOAS).

Conclusión

La integración de OWL en JC3IEDM dentro de una arquitectura de aprendizaje federado ofrece una solución robusta para los desafíos de interoperabilidad y privacidad en sistemas distribuidos de datos, como aquellos en el ámbito de la defensa y seguridad. Al enriquecer JC3IEDM con OWL, es posible no solo representar los datos con mayor flexibilidad y precisión semántica, sino también facilitar la colaboración entre entidades sin comprometer la privacidad de la información. Esta propuesta permite que los datos se mantengan localizados mientras se entrena un modelo global, lo cual optimiza la eficiencia y protege la confidencialidad de los datos en una arquitectura federada.

Para implementar esta integración, se propone desarrollar una arquitectura de aprendizaje federado que incorpore una capa semántica basada en OWL, aplicada a JC3IEDM. Esto implica crear una ontología que expanda y enriquezca las entidades y relaciones de JC3IEDM, definiendo con precisión las características y relaciones semánticas necesarias para los datos en un entorno federado con UAVs

Bibliografía

- [1] “Multi-Domain Combat Cloud - Joint Air Power Competence Centre.” Accessed: Feb. 20, 2024. [Online]. Available: <https://www.japcc.org/essays/multi-domain-combat-cloud/>
- [2] K. Demertzis, P. Kikiras, C. Skianis, K. Rantos, L. Iliadis, and G. Stamoulis, “Federated Auto Meta-Ensemble Learning Framework for AI-Enabled Military Operations,” *Electronics*, vol. 12, no. 2, Art. no. 2, Jan. 2023, doi: 10.3390/electronics12020430.
- [3] “The Joint C3 Information Exchange Data Model Overview”. Accessed: Nov.1, 2024, Available: <https://brainsciencesjournal.org/2007.JC3IEDM.pdf>
- [4] Lasschuyt, E., Marcel Van Hekken, M., Treurniet, M. W., & Visser, M. M. (s/f). “How to Make an Effective Information Exchange Data Model or The Good and Bad Aspects of.” Available: <https://apps.dtic.mil/sti/pdfs/ADA469736.pdf>
- [5] Christopher J. Matheus¹ and Brian Ulicny “On the Automatic Generation of an OWL Ontology based on the Joint C3 Information Exchange Data Model”. Available: http://www.dodccrp.org/events/12th_ICCRTS/CD/html/papers/149.pdf