NASA/TM–20210000030

# An Ontology-Based Virtual Orrery

*D.A. O'Neil*
*Marshall Space Flight Center, Huntsville, Alabama*

*R.J. Rovetto*
*NASA Datanauts, New York, New York*

***January 2021***

## The NASA STI Program…in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and mission, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results…even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>

- E-mail your question via the Internet to <help@sti.nasa.gov>

- Phone the NASA STI Help Desk at 757–864–9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681–2199, USA

NASA/TM–20210000030

# An Ontology-Based Virtual Orrery

*D.A. O'Neil*
*Marshall Space Flight Center, Huntsville, Alabama*

*R.J. Rovetto*
*NASA Datanauts, New York, New York*

National Aeronautics and
Space Administration

Marshall Space Flight Center • Huntsville, Alabama 35812

***January 2021***

## Acknowledgments

This work was made possible by the Datanauts activity sponsored by the NASA Headquarters Office of the Chief Information Officer. Authors of this document would like to thank Beth Beck for establishing the Datanauts program, Lori Parker, the former program manager, and David Meza, the current program manager. Also, thanks go to former Datanauts community managers Elyssa Dole and Veronica 'Ronnie' Phillips, as well as the current community manager, Claire Little. Additional thanks go to Jared Austin and Troy Farsoun for editing this Technical Memorandum. This work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

Available from:

NASA STI Information Desk
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681–2199, USA
757–864–9658

This report is also available in electronic form at
<http://www.sti.nasa.gov>

## TABLE OF CONTENTS

**TABLE OF CONTENTS (Continued)**

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF ACRONYMS

| | |
|---|---|
| AI | artificial intelligence |
| API | Application Programming Interface |
| CLIF | Common Logic Interchange Format |
| COSPAR | Committee on Space Research |
| CSV | comma-separated value |
| DAML | DARPA Agent Markup Language |
| DL | description logic |
| GUID | Globally Unique Identifiers |
| HTML | HyperText Markup Language |
| IRI | internationalized resource identifier |
| JSON | JavaScript object notation |
| JSON-LD | JavaScript object notation linked data |
| KIF | Knowledge Interchange Format |
| LEO | low Earth orbit |
| NEO | near Earth objects |
| NORAD | North American Aerospace Defense Command |
| ODO | Orbital Debris Ontology |
| OWL | Web Ontology Language |
| PHA | potentially hazardous asteroid |
| RAAN | right ascension of the ascending node |

## LIST OF ACRONYMS (Continued)

| | |
|---|---|
| RDF | resource description framework |
| SAR | search and rescue |
| SPARQL | SPARQL Protocol and RDF Query Language |
| SQWRL | Semantic Query Web Rule Language |
| SSA | space situational awareness |
| SSAO | space situational awareness ontology |
| SWRL | Semantic Web Rule Language |
| TLE | two-line element set |
| TM | Technical Memorandum |
| UML | Unified Modeling Language |
| URI | uniform resource identifier |
| WMVS | Web-Based Mission Visualization System |

## NOMENCLATURE

| | |
|---|---|
| $M$ | mean anomaly |
| $t$ | time since epoch, where epoch is the time for the initial value of the mean anomaly |
| $\alpha$ | semi-major axis |
| $\mu$ | standard gravitational parameter |
| $\eta$ | mean motion |

TECHNICAL MEMORANDUM

# AN ONTOLOGY-BASED VIRTUAL ORRERY

## 1. INTRODUCTION

Modeling complex systems, integrating disparate data, defining classification rules, and structuring data files for software applications can benefit from an ontology. An ontology formally defines a structured set of classes of things, relationships among them, and their attributes. Ontologies define semantically-rich taxonomies to structure unstructured data and facilitate standardization of structured data within a domain of knowledge. Inference rules and queries enable the production of additional information and reports. A few examples of applications for ontologies include system models, knowledge bases, expert systems, and data file generators. Structured text formats enable storage, exchange, and reuse of ontologies; for example, metadata for several thousand NASA datasets are expressed in the compact JavaScript Object Notation (JSON) Linked Data (JSON-LD) format.

The remainder of this Technical Memorandum (TM) is arranged as follows: Section 2 explains ontologies and provides a brief overview of some ontology editors. Section 3 provides an overview of the Space Situational Awareness Ontology (SSAO), an ontology project available for the astronautical community, which was successfully applied in this application. Section 4 explains JSON-LD. Section 5 describes a procedure for developing webpages with data visualizations generated from an ontology, such as the SSAO. Section 6 presents a JavaScript-based demonstration of a virtual solar system model that reads planetary orbital parameters from a JSON-LD file generated from the SSAO. Section 7 explains how to configure the Protégé ontology editor so that instantiated objects will receive names from imported data. Section 8 presents an R language tutorial for generating a webpage with a table and an interactive 3D data visualization using a JSON-LD data file that was generated from classes in the SSAO. Section 9 offers a conclusion of the primary benefits of the taxonomy and terminology defined within the SSAO.

## 2. WHAT IS AN ONTOLOGY?

Ontology has origins in philosophy as inquiry into the most general aspects of the world. It is concerned with the nature of existence, what exists, what can exist, and how they are related. Ontological accounts of the world, or some specific topic area, often involve developing categories and interrelationships to systematically represent various kinds of things. Formal ontology uses logics, such as first-order predicate calculus, to precisely describe them. Similar concepts and activities are found in computational disciplines.

Within information and computer science, an ontology is a formal model or representation of the types of properties, objects, events, processes, and relationships that comprise a universal set in a specified domain of knowledge. Tom Gruber defined an ontology as "an explicit specification of a conceptualization,"[1] which involves formalizing categorical concepts, specific individuals instantiating those categories, as well as background and implicit knowledge. Ontologies have also been called 'semantic models' and 'knowledge models,' and they overlap with conceptual modeling and similar activities. They are a kind of knowledge organization system, alongside thesauri, controlled vocabularies, and topic maps. While a data model may express the structure of data, ontologies (or ontological models) express the meaning of data. This makes ontologies a semantic layer on databases.

The key role of ontologies with respect to database systems is to specify a modeling representation at a level of abstraction above specific database designs, so that data can be exported, translated, queried, and unified across independently developed systems and services.

Ontology engineering is interdisciplinary and is related to knowledge representation and reasoning, a branch of artificial intelligence (AI). It is also associated with semantic web technologies, informatics, so-called 'big data,' linked data, knowledge graphs, data science, and knowledge management. Knowledge-based approaches, such as ontology development, have been applied in astronomy, geography, and space operations, among other disciplines. Examples of NASA-developed semantic resources include:

- Semantic Web for Earth and Environmental Terminology (SWEET) ontologies[2]
- NASA Exploration Initiative Ontology Models (NExIOM)[3]
- NASA Air Traffic Management Ontology[4]
- NASA Metadata catalog[5]

Examples of ontologies being developed by Robert Rovetto include:

- The Orbital Space Environment Domain Reference Ontology[6,7]
- The Space Situational Awareness Ontology (SSAO)[8,9]
- The Orbital Debris Ontology (ODO)[10]

- The Orbit Ontology[11]
- The Spacecraft Ontology
- The Orbit Data Format/Message Ontologies
- The Union of Concerned Scientists Satellite Ontology[11]
- The Space Traffic Management Ontology
- The Space Policy & Law Ontology.

The second group of ontologies is part of a project vision described online at <https://purl.org/space-ontology>.[12] A recent paper[13] provides an overview with potential applications. These ontologies, e.g., the SSAO, are intended in part to provide metadata and ontological domain models for the astronautics community at large. The domain concepts expressed by defined terms may link and tag data of distinct formats and data sources such as NASA datasets, Space-Track.org, CelesTrak, spacecraft operator datasets, academia projects with satellite data, etc. This facilitates semantic interoperability and data queries. Additional information on other knowledge-based approaches to space data is discussed in reference.[14] The project is open to contributors.

Ontologies can be interconnected, extended, reused, and mapped to others. Potential use cases include: semantic annotations (tags) for resources on the web and external databases, knowledge models, expert systems, data integration, information fusion, data search and retrieval, and knowledge discovery via automated inference. Clearly annotating data elements, for example, affords more advanced capabilities such as machine learning. An ontology can represent domain knowledge by asserting axioms, rules, and specifying taxonomies comprised of classes/categories, interrelationships among the classes, individuals, and their properties. Metadata associated with these classes and relations can be included for human readability as well. An example, specified with constructs from the OWL knowledge representation language[15] and the Protégé ontology editor,[16] is as follows. The following categories in single quotation marks are capitalized with underscores, and relations have dashes separating lower-camel-cased words.

'Spacecraft' is a class of things. 'Earth_Observation_Satellite' is a subclass. Subclasses inherit the properties of their parent or superclass; this facilitates automated reasoning. So-called OWL 'Data properties' are relations between individuals and a specific value. Data properties associated with Spacecraft include: length (or has-Length), height, width, mass, launch-Date, etc. A particular thing or object (rather than a class), known as a 'named individual,' instantiates a class. 'Landsat 7,[17] for example, is a named individual that instantiates the class Earth_Observation_Satellite. Attributes of an object are values assigned to related data properties. Values can have data types, such as Integer, String (e.g., xsd:string), decimal, and float. Attributes together with datatypes (of a given data property) for the Landsat 7 individual include: 4.04 m, 2.74 m, 2.74 m, 2,200 kg, April 15, 1999.

An inference engine (or reasoner) can assign named individuals to classes. It can also generate reports based upon semantic rules and queries that search for specified attributes. Inference rules can generate additional information; for example, with the appropriate rules, axioms and properties, if Alice has-Child Clara and has-Parent David and Bob has-Parent David then the reasoner will automatically infer Clara has-Uncle Bob.

An inference rule could calculate volume based upon length, height, and width, while another rule could assign a value of small, medium, or large to an object's size property based upon volume. This automated reasoning can be applied to orbital parameters and other data in astronomy, space situational awareness (SSA), and other disciplines. Consider space mission design and implementation.

Designing, analyzing, and presenting concepts associated with a space mission involve several knowledge domains. A few of these domains include the space environment, spacecraft architecture, payload descriptions, sequences of events for the mission, and data structures for data visualization. In the past, these data were stored with data files, documents, and charts within presentations. By designing interoperable ontologies and software applications that use those ontologies, future conceptual design, analysis, and visualization activities can ensure that all of the input data are captured in a way that facilitates repeatability, reusability, and improves documentation, reporting, and change management.

### 2.1 Structured Terminology

Although some ontologies may be a flat set of terms, they are often composed of structured hierarchical terminologies, such as taxonomies, whose terms are given a formally-specified semantics. This includes category terms (labels for the class construct in OWL), and relational terms (for OWL object/data properties). These taxonomies are organized using structuring relations such as Class Subsumption (Parent-Child relationships), sometimes called the 'is-a' or 'is-a-kind-of' relation, and partonomic relationships (parthood), such as 'part-of'.

Each construct in an ontology can be expressed by a term label. These terms composing the ontology's vocabulary can be given a natural language definition for human readability. The ontology's formal definitions uses a knowledge representation (or ontology) language, which makes it machine-readable (computable). Development tools often provide a choice of implementation languages or serializations.

Examples of these formal languages[18] are the Resource Description Format (RDF),[19] Web Ontology Language (OWL),[15] Knowledge Interchange Format (KIF), and the Common Logic Interchange Format (CLIF).[20] Together with the logical model of an ontology, this affords automated reasoning, consistency checking, classification, and queries (e.g., using SPARQL,[21] SWRL[22]) over data tagged with ontology terms.

Ontology constructs, or their terms, are used as semantic annotations for resources on the web and for datasets housed in databases. They can tag various data elements. Definitions associate semantic content (or meaning) to that data. Thus, ontologies explicitly convey the meaning of data that might otherwise have been implicit, scattered across various documents in an organization, or confined to the minds of database administrators, programmers, or subject matter experts. This semantic layer is transferable and usable across siloed and disparate databases. Ontologies potentially offer shareable models (and semantics) for data of a common subject matter. This can facilitate data sharing and interoperability.

While a database may store data on individual objects (e.g., the International Space Station), an ontology can represent both individuals and categories (e.g., 'Space_Station') to classify and characterize them. It captures both generic and specific statements and knowledge.

By adding logical constraints to the semantics, an ontology can capture generic and universal domain knowledge to be shared across databases (or among ontologies). Formal constraints narrow the intended meaning of classes and interrelationships and can take the form of rules, logical axioms, and other expressions. Conceptually, an ontology can be as expressive and detailed as needed. However, the selected implementation language, and tools, may impose expressive limitations. In this way, an ontology is more than a structure terminology. All of this can be developed in an ontology or knowledge modeling editor tool.

### 2.2 Ontology Editors

Ontology editing and development tools enable the integration of multiple ontologies and publishing data files that are tailored to specific applications. Concepts and terms of an ontology can have both human and computer readability descriptions. Examples of the latter include: DARPA Agent Markup Language (DAML),[23] RDF,[19] KIF, OWL,[15] and CLIF.[20] Zhang and Miller evaluated the performance of a few languages and found that RDF is more limited as compared with OWL for representing real-world information; OWL is flexible, expressive, accessible by computers, and readable by humans.[18] OWL limitations sometimes requires ad hoc workarounds. KIF and CLIF, however, are more expressive than OWL, allowing for richer semantics. Ontologies can also be built using model-based systems engineering tools such as MagicDraw, the Systems Modeling Language (SysML), the Unified Modeling language (UML), and Object Role Model (ORM). These and other implementation languages are offered as options in some modeling tools.

Editor software thereby provides a capability to create, edit, merge, and publish ontologies and other semantic systems in a variety of formats. Typical features and functions of editors include a graphical user interface with a hierarchical list of the taxonomy, an interactive graphical representation of the ontology, an inference engine, and a text editor for inference rules and queries. Example semantic editor tools include the following:

- TopBraid by TopQuadrant[24]
- OntoStudio by Semafora[25]
- Apollo by Knowledge Media Institute[26]
- Hozo Ontology Editor by Osaka University[27]
- OWLGred by the University of Latvia[28]
- Fluent Editor for PC by Cognitum[29]
- Protege by Stanford Center for Biomedical Informatics Research[16]
- PoolParty Semantic Suite[30]

Emhimed Alatrish compared functions and features of the first three ontology editors on the preceding list.[31] Robert Rovetto revised the English version of the Hozo Ontology Editor Operating Manual.[32] Diagrams drawn with Hozo generate an ontology. A unique feature of

OWLGred is that it generates UML diagrams that represent an ontology.[28] Fluent Editor enables development of ontologies with controlled English. The Stanford Center for Biomedical Informatics Research developed a free ontology editor named Protégé. A feature of the Protégé ontology editor that enables the development of ontology-based web apps is that it can save files in the JavaScript Object Notation (JSON) Linked Data (JSON-LD) format.[33] The next section explains JSON-LD.[34] Following this, the SSA ontology is described; it forms the ontology component of a demonstration of the ontological application at hand.

## 3. JAVASCRIPT OBJECT NOTATION LINKED-DATA

JSON is a standardized data interchange text format for messages or storage that is readable by humans and software.[34] The JavaScript JSON.parse() command accesses a JSON data file or message and stores the data in a variable.[33] In 2014, the World Wide Web Consortium (W3C) recommended a JSON-based serialization of linked data called JSON-LD.[35]

With JSON-LD, ontologists and data scientists can structure data as a labeled directed graph. Terms that define a graph are nodes and arrows that link them (also called 'edges'). An internationalized resource identifier (IRI) is a globally unique named path that may or may not lead to a web-page; this term is slowly replacing the terms 'uniform resource identifier' (URI) and 'uniform resource locator' (URL). Nodes specify a data value, list, or an IRI. Edges link a pair of nodes and have a direction, IRI, and label property. The graph form may have the RDF serialization, whose format is subject-predicate-object. A statement or piece of domain knowledge asserted in an ontology often takes this form. An example is: Landsat7 has Mass 2,200 kg.

Section 6 of the JSON-LD recommendation describes three forms of a JSON-LD document: expanded, compacted, and flattened. The expanded format is rather verbose because every object includes an IRI. In the compact form, the document developer defines a context for the IRIs. Flattening a JSON-LD document collects all properties of a node into a single JSON object and labels all blank nodes with identifiers, e.g., "@id":"_:b0". According to the recommendation, a flattened document "may drastically simplify" processing code. The following sections explain how JSON-LD can be used in ontology development applications.

## 4. OVERVIEW OF THE SPACE SITUATIONAL AWARENESS ONTOLOGY

The Space Situational Awareness Ontology (SSAO)[8,36] is an ontology representing the objects, properties, events, concepts, knowledge and data of the space situational awareness domain. It is part of a broader ontology vision of a suite of ontologies for the spaceflight community, a living project tentatively called the Orbital Space Environment Domain Reference Ontology[6,7] project, or the Orbital Space Knowledge Modeling project. It aims for knowledge representation and reasoning for astronautics and astronomy.

This astronautical knowledge modeling suite aims to provide a set of modular domain or reference ontologies to support spaceflight and SSA. It is presently described at <https://purl.org/space-ontology>[12] and associated GitHub pages via https://github.com/rrovetto/.[30] All ontologies are subject to revision, including the scope, titles, identifiers, and content. Ontology files, such as the SSAO OWL file, are tentatively expected to be accessible via permanent IRIs such as <https://purl.org/space-ontology/ssao>.[36] Visit the websites and contact the author for further information. Partners and formal opportunities to fully realize the envisioned system is desired. The envisioned system will ideally include an ontology-annotated 3D visualization of the orbital space environment.

The ontology of SSA and desiderata for a computational SSA ontology were discussed by Robert J. Rovetto and T.S. Kelso in 2016.[8] It is based on a 2015 paper[10] that proposed alleviating the orbital debris problem by using ontology to facilitate data sharing and interoperability in the community. The ODO mentioned in the 2015 paper is a related in-progress domain ontology.[10] See <https://purl.org/space-ontology/odo>.[37] It is relevant for space agency debris programs, and the project hopes to be used by and developed with offices such as NASA's Orbital Debris Program Office.[38] Other ontologies in the suite include ontologies for spacecraft, their parts, and their operations, as well as orbital events, activities, space missions, etc.

The scope of these ontologies covers entities in the above-mentioned disciplines: astronautics, orbital debris, SSA, space traffic management, spaceflight missions, orbital dynamics, spacecraft operations, etc. They will ontologically represent and classify: orbits, orbital parameters, orbital debris objects; sensor devices; observational and predicted data with their sources and formats; surveillance, detection, tracking and propagation activities; spacecraft and their subsystems; spaceflight maneuvers; space missions; participating agents in the space domain; etc. Astronomy necessarily overlaps with these target domains, and knowledge thereof will be ontologically modeled. Existing astronomical ontologies may be interconnected with the SSAO and sibling ontologies, importing or mapping terms. Existing ontologies by NASA or elsewhere can also be interconnected.

The ontologies are composed of formally defined classes, properties (or relations), and individuals, each of which is given a human-readable label (a term). Sample classes, presented here in camel case with no spaces, include: Orbit, HeliocentricOrbit, LowEarthOrbit, OrbitalEccentricity, SpaceSurveillanceNetwork, SpaceObservationProcess, SpaceObject, OrbitalObject, ResidentSpaceObject, AstronauticalSystem, Spacecraft, ArtificialSatellite, OrbitalDebrisObject, Two-LineElementSet, SpacecraftComponent, ArtificialSatellitePart, CentralBody, SensorObservation, etc.

Sample relations include: hasOrbitalEccentricity, hasAttitudeState, hasOrbit, hasLaunchVehicle, hasProbabilityOfCollision, hasTimeOfClosestApproach, isTrackedBy, hasCentralBody, hasDataSource, hasOperationalStatus, isOrbitedBy (with inverse, orbits), etc.

Classes and relations will be grouped into appropriate ontology modules for more efficient (re)use and computation. They can then be imported into a master ontology file to capture the overall domain or used separately. For example, categories of orbits will be located in an Orbit Ontology, which can then be imported into the SSAO file or into your own ontology file. This modular architecture—interoperable ontologies covering part of the intended domain—should allow each module to be shared and applied across organizations and system. Users may select specific ontologies of interest. The classes, properties, and relations are defined at a sufficient degree of abstraction to tag or annotate similar data or organization content from different enterprises or agencies. This semantic annotation affords semantic interoperability by making explicit which data elements have a common meaning and providing a common terminology for disparate systems and groups.

The ontologies can semantically describe the generic properties of an orbital regime, and the specific property values of an orbit at a particular time. Inference rules will allow automated reasoning on orbital data from distinct data sources. Axioms, constraints, SPARQL and SQWRL rules will be declared to facilitate database queries and capture domain knowledge. These reasoning functionalities will afford such things as identifying categories of debris, orbits, hazard assessments, etc.

The ontologies will be reusable, as well as extendable, by more specific ontologies. They also will be interoperable with more abstract (generic or foundational) ontologies. They will be implemented in more than one knowledge representation language for wider applicability and greater expressivity.

The formal methods employed, combined with philosophical, lexical, logical, and conceptual analysis, have conceptual and practical benefits. Practical goals for the SSAO and the entire space domain ontology suite include, but are not limited to:

- Knowledge representation and reasoning of fundamental orbital concepts, SSA data, etc.
- The classification of orbital debris and other space objects.
- Data-sharing and interoperability among federated databases; information fusion.
- Offering metadata elements, and a set of terms, to link data and connect systems.

- Terminology harmonization and standardization.
- Improving exiting definitions and terminological standards.
- Artificial intelligence support
- Ontology-based visualizations
- Helping resolve open terminological and conceptual questions in space policy and law.

In all, this will provide a unifying model and formal terminology capturing sharable knowledge. By developing these formal ontological resources in a systematic, sound, and neutral manner, they may serve as standards for the community. The critical analysis can inform improvements to existing terminologies in data and space communities, alike. Moreover, participation and community involvement will help ensure development and quality assurance.

The SSAO contains concepts relevant and applicable to various applications and information systems in the space domain. In the following sections, we demonstrate an example of a successful application of the SSAO, an application at the intersection of web-based platforms and 3D visualizations. The SSAO contains classes and relations relevant for this involves the use of JSON-LD and R. We begin with outlining ontology development procedures using Protégé and its modeling constructs.

## 5. AN ONTOLOGY APPLICATION DEVELOPMENT PROCEDURE

Modeling complex systems or knowledge domains with an ontology editor (here, Protégé), involves specifying a hierarchy of classes of things, defining data properties that specify attributes, and defining object properties that describe relationships among things. An agreed-upon terminology can help structure unstructured data and facilitate standardization of structured data and its semantics (meaning).

As explained in a previous section, named individuals, sometimes known as 'objects' in Protégé, instantiate classes. An object's attributes are values assigned to data properties. Creating a lot of objects and specifying attributes within an ontology editor can become tedious and time-consuming. Fortunately, version five of Protégé includes a function called "Create Axioms from Excel workbook," which enables object instantiation and attribute assignments from imported data. This section describes two general procedures, and the three tutorials that follow this section apply to these procedures. One procedure creates an ontology, and the other imports an existing ontology or parts of it. Figure 1 depicts a procedure for creating either a knowledge base or an expert system using Protégé. Steps in this procedure include:

(1) Assert classes, data properties, and object properties: This step produces the terminology that constitutes a basic ontology. If the objective is to develop a knowledge base, consider questions people want to ask about the data. If the objective is to develop an expert system, analyze the type of data and how it can be processed to generate additional data.

(2) Import data into ontology editor: Transformation rules for importing data into Protégé can be written. They can assign instances to classes and assign object attributes by mapping columns from an Excel spreadsheet to classes and data properties in the ontology.

(3) Write inference rules and queries: This step could occur before the previous step; however, it may help to see the data within the context of the ontology. Protégé supports the Semantic Web Rule Language (SWRL) for inference rules.

(4) Infer object classification, relationships, or attributes: Executing inference rules generates additional information. The section, "What is an Ontology?" provided a high-level example of inferring an uncle relationship from parent child relationships. Another example in that section described the inference of volume from the dimension's length, height and width.

(5) Generate CSV reports via queries: Queries in Protégé can be written in Description Logic (DL), SPARQL Protocol and RDF Query Language (SPARQL), and the Sematic Query Web Rule Language (SQWRL). Queries can generate reports. Protégé can save query-generated reports as a comma separated value (CSV), which can be opened in Excel.


Figure 1. Example workflow of importing and manipulating data in an ontology.

Figure 2 depicts a procedure for generating a hierarchically structured data file for use in custom-developed software. This procedure assumes that the taxonomy of an ontology has already been developed. Steps in this procedure include:

(1) Import an ontology, or selected classes from an ontology, into the editor: Classes within a selected ontology, such as the Orbit class from the SSAO, defines an orbit and attributes define the orbital parameters. In the workflow diagram, individuals were instantiated for the inner planets and the orbital parameter attributes were populated with data about the planets orbits.

(2) Export a JSON-LD file: Protégé provides a function to save a JSON-LD file. An important feature of Protégé's JSON-LD save function is that only the instantiated classes are exported. This feature enables an application developer to integrate multiple ontologies and create a custom data file with only the necessary data in the expanded format. A JSON-LD file exported from Protégé is in the expanded format, which is verbose and can be difficult to read.

(3) Convert the JSON-LD file from expanded to compact format: The group that developed the JSON-LD standard provides an example conversion utility at <http://json-ld.org>.[39] Converting from the expanded to the compact format makes the JSON-LD easier to read because the @context section specifies brief terms that are substituted for the URI. More importantly, substituting short terms eliminates the @ sign, which enables computer languages to read the data as if it were a standard JSON file.

(4) Develop a web app to use the JSON-LD data file: JavaScript includes Parse function for reading a JSON file. Computer languages R, Python, and others have code libraries that can read JSON files, which work with compact JSON-LD. In JavaScript, parsing a JSON file imports records into a variable so the code can access data fields by referencing. variablename.fieldname. In R, data fields can be accessed with variablename$fieldname.

(5) Implement code to use the parsed data: The web app can format the parsed as a table, fill in fields within paragraphs, or feed equations that generate data for a visualization.


Figure 2. Example workflow involving an ontology to generate a data file.

Figure 3 presents process for developing ontology driven web apps. In the example presented in figure 3, an early version of the SSAO was imported (step 1), a JSON-LD file was the output (step 2), which was converted to a compact format (step 3). This in turn was read and built into a HTML-based table and rendered on the web (steps 4 to 5).

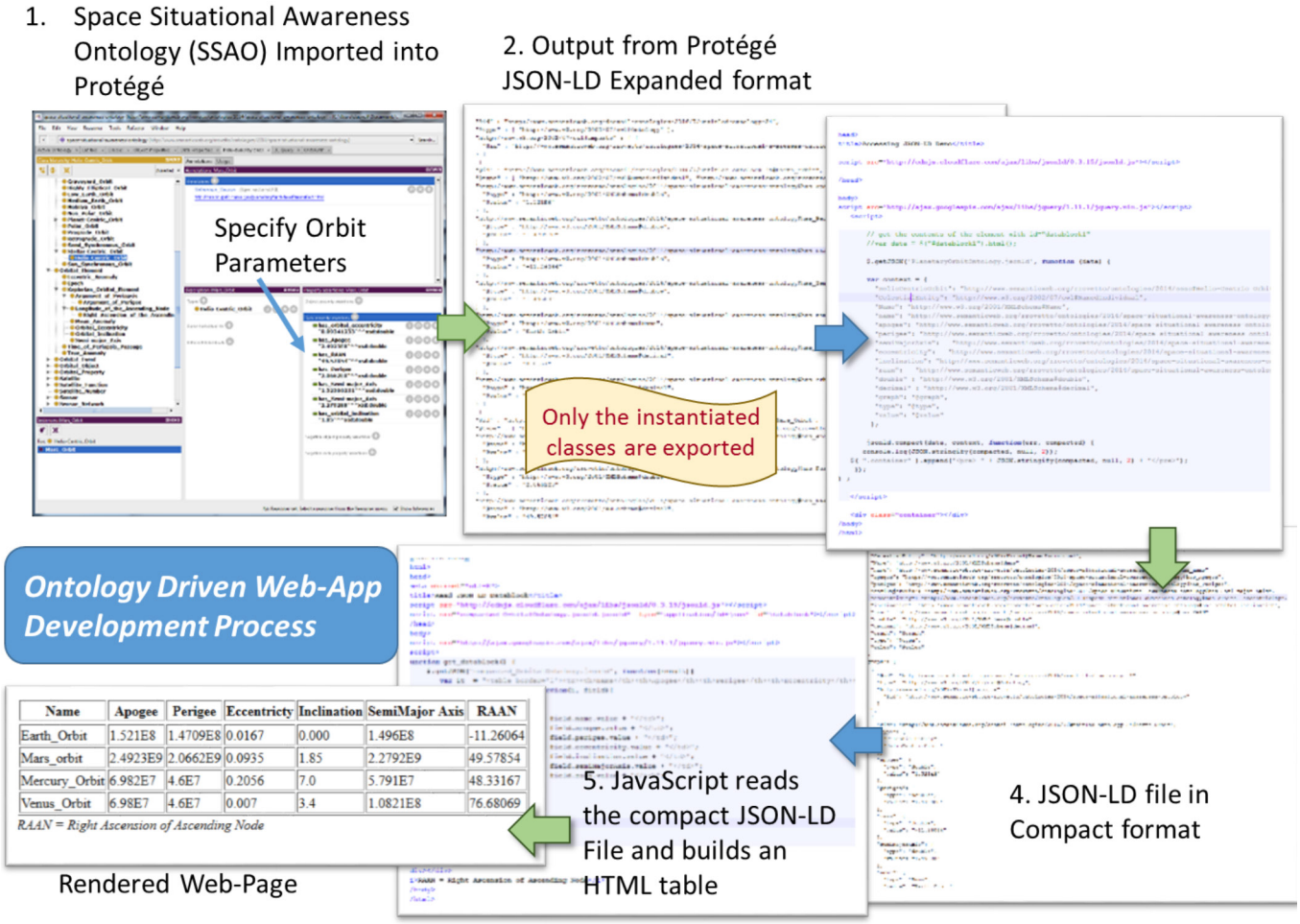



Figure 3. Process for developing ontology-driven web apps.

Some classes within the SSAO specify types of orbit. Orbital property attributes define the orbital parameters. In the workflow diagram, individuals were instantiated for the inner planets and the orbital parameter attributes were populated with data about the planet's orbits. The following section presents a demonstration of an ontology-oriented orrery developed with the above workflow.

## 6. DEMONSTRATION: AN ONTOLOGY-BASED ORRERY

Since Charles Boyle, the Fourth Earl of Orrery, received a mechanical model of the Solar System in 1704, people have referred to mechanical solar system models as orreries.[40] With WebGL, it is possible to create interactive 3D web apps that present a virtual orrery. Using the procedure described in section five, together with the SSAO, an ontology-based orrery was developed. This demonstrates a dynamic visual application of ontologies, specifically the SSAO.

Named individuals representing the inner planets were associated with instantiated orbital parameters and exported as a JSON-LD data file. JavaScript code reads orbital parameters from a compact JSON-LD file and feeds the data to trajectory propagator.

Figure 4 depicts the 3D visualization of the ontology-oriented orrery. Planets orbit the Sun along the traced orbits. Interactive features of this web-based orrery include pan, zoom, and rotate. The code library for rendering is Three.js.

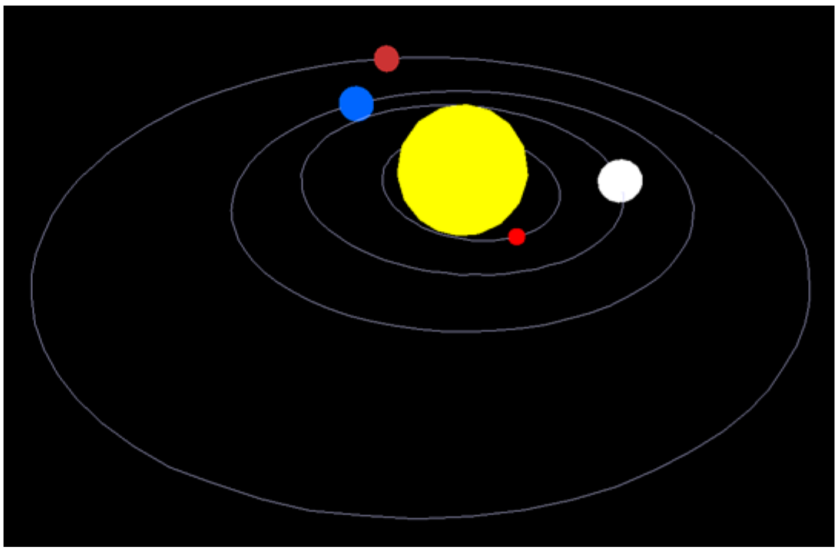


Figure 4. A screenshot of the ontology-based orrery.

The expanded and compacted formats are discussed in sections 6.17 and 6.18 of the WC3 JSON-LD document.[33] The reader may also consult the example in the Digital Bazaar GitHub Readme page.[41] Development of the ontology-oriented orrery followed that procedure to produce the data file for a JavaScript function that renders the orrery. Figure 5 presents JavaScript code that uses the Three.js code library to establishe a virtual camera and creates the spheres that represent the Sun and the orbiting planets.

```
function init() {
   camera = new THREE.PerspectiveCamera(
            75, window.innerWidth / window.innerHeight,
            0.1, 1000 );

   renderer = new THREE.WebGLRenderer();
            renderer.setSize( window.innerWidth, window.innerHeight );
            document.body.appendChild( renderer.domElement );
var j = 0 ;
var geometry = new THREE.SphereGeometry( 0.6, 16, 16 );

// Sun
var material = new THREE.MeshBasicMaterial({color:0xFFFF00,
                  wireframe: false});
      sphere = new THREE.Mesh( geometry, material );
                  scene.add( sphere );

for (i = 0; i < 4; i++) {      // Add the inner planets.
  geometry = new THREE.SphereGeometry( objSizes[j], 16, 16 );
  material = new THREE.MeshBasicMaterial({color:objColors[j],
                wireframe: false});
        sphere = new THREE.Mesh( geometry, material );
        sphere.name = objNames[i] ;
           scene.add( sphere );
      j = j + 1 ; }
           camera.position.z = 5;
 }
```

Figure 5. JavaScript code using the ThreeJS library to construct an orrery.

Tom Logson's book *Orbital Mechanics: Theory and Applications*[42] explains the Keplerian parameters, relationships among positions, known as anomalies, on an ellipse and a circle. It also provides a flow chart for an orbital propagator. Code presented in this section was inspired by those explanations and flow chart.

Figure 6 presents the anomalies, which are briefly explained in this paragraph. Considering the second law of planetary motion, an orbiting body changes velocity as it traverses about an elliptical orbit. In circular orbits, an orbiting body has a constant velocity. Superimposing a circle over an elliptical orbit provides a virtual clock wherein the position of a virtual sweep hand moves by a constant incremental distance at a regular interval; this numeric value is called the mean anomaly[42] $M = \eta t$, where:

$$\eta = \sqrt{\frac{\mu}{\alpha^3}} \quad . \tag{1}$$

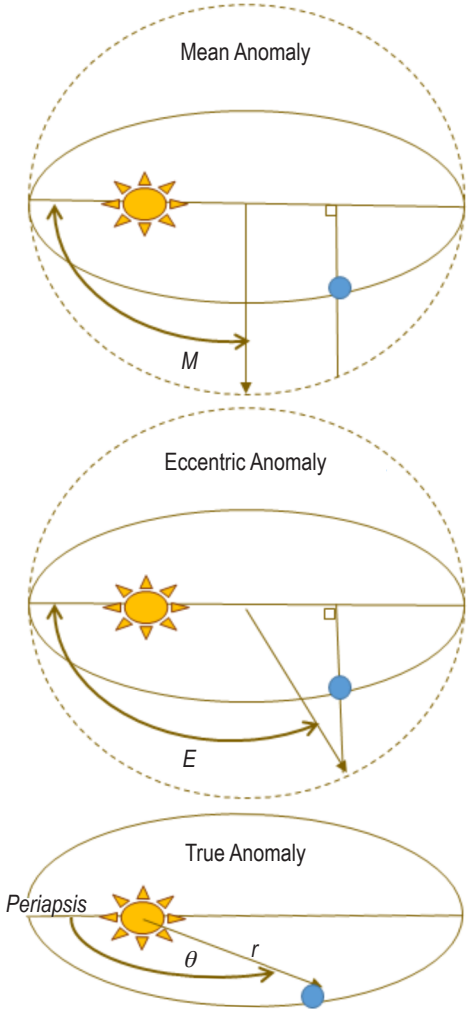



Figure 6. The anomalies.

The variable depends upon the standard gravitational parameter of the primary celestial body (the orbited body); for Earth, $\eta = 3.986004418e^5$ km$^3$/s$^2$, and $\alpha$ is the semi-major axis. The mean anomaly is the virtual sweep hand that extends from the center of the ellipse to the superimposed circle. Projecting a perpendicular line from the central axis through the orbiting object on the ellipse to the superimposed circle provides an end point for a line that originates at the center of the ellipse as depicted in the figure. The eccentric anomaly is the angle between a tangential point of the circle and ellipse to the projected end point. The true anomaly is the ever-changing radius from the primary body located at a focus of the ellipse to the calculated position of orbiting body on the ellipse. The position of orbiting body is projected back from the superimposed circle. Figure 7 depicts a flow chart for an orbital propagator implemented by the JavaScript code that follows the figure.

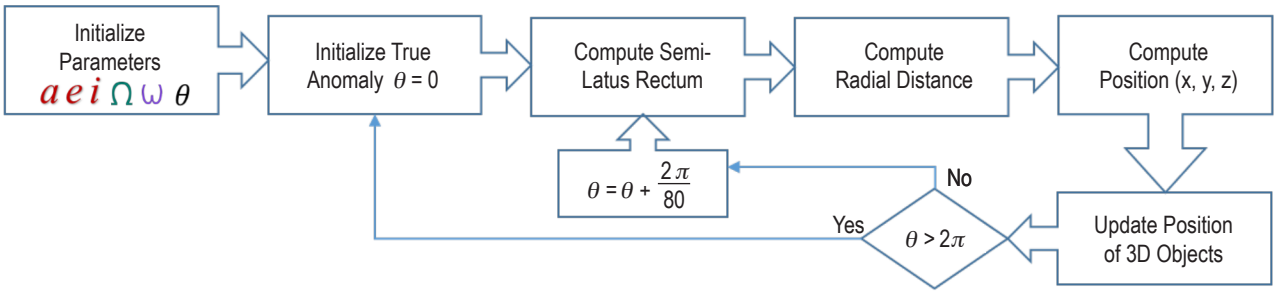



Figure 7. A flow chart for an orbital propagator.

Figure 8 presents ECMAScript, also known as JavaScript, code that was developed in accordance with the flow chart depicted in figure 7. Functions presented in figure 8 convert values from the mean anomaly to the eccentric anomaly, the eccentric anomaly to the true anomaly, and the true anomaly to the eccentric anomaly.

```
      function meanToEccentricAnomaly (e, M) {
            // Determines eccentric anomaly, E from a given
            // mean anomaly, M and eccentricty, e.
            // Performs a simple Newton-Raphson iteration
      // Code derived from Matlab scripts written by Richard Rieber,1/23/2005
      // http://www.mathworks.com/matlabcentral/fileexchange/6779-calce-m
      var tol = 0.0001;  // tolerance
      var eAo = M;       // initialize eccentric anomaly with mean anomaly
      var ratio = 1;     // set ratio higher than the tolerance

      while (Math.abs(ratio) > tol) {
            var f_E = eAo - e * Math.sin(eAo) - M;
            var f_Eprime = 1 - e * Math.cos(eAo);
            ratio = f_E / f_Eprime;
            if (Math.abs(ratio) > tol) {
                  eAo = eAo - ratio;
        }
            else
             eccentricAnomaly = eAo;
      }
            return eccentricAnomaly
     }

function eccentricToTrueAnomaly(e, E) {
 //  http://mmae.iit.edu/~mpeet/Classes/MMAE441/Spacecraft/441Lecture19.pdf
 //   slide 8
   var trueAnomaly = 2 * Math.atan(Math.sqrt((1+e)/(1-e))*
       Math.tan(E/2));
      return trueAnomaly
}

 function trueToEccentricAnomaly(e,f) {
// http://mmae.iit.edu/~mpeet/Classes/MMAE441/Spacecraft/441Lecture19.pdf
// slide 7
 var eccentricAnomaly = 2* Math.atan(Math.sqrt((1-e)/(1+e))*
      Math.tan(f/2));
      return eccentricAnomaly ;
   }
```

Figure 8. JavaScript function to convert the values among the anomalies.

Figure 9 presents JavaScript code for an orbital propagator. Within the context of the flow chart depicted in figure 7, the orbital propagator updates the position of the 3D objects. Figure 10 presents a portion of a compact version of the JSON-LD file that was exported from the SSAO. This portion of the file specifies the Keplerian parameters for Earth. To improve readability and conserve space, a long URI was replaced the short URI <http://www.purl.org/ssao/>.[36] Note that the URIs in figure 10 for terms in the SSAO may not be active at the time of this publication. New URIs/IRIs will be generated for the entire vocabulary as development continues.

```
Trajectory.prototype.propagate = function(uA){
      // Purpose: Determine a position on an orbital trajectory based upon
      // a true anomaly value.

      var pos = [] ;                        // Position
      var theta = uA;                       // Update true anomaly.
      var smA = this.smA;                   // Semi-major Axis
      var oI =  this.oI ;                   // Orbital Inclination
      var aP = this.aP ;                    // Argument of perigee
      var oE = this.oE;                     // Orbital eccentricity
      var aN = this.aN ;                    // Ascending Node
      var sLR = smA * (1 - oE^2) ;             // Compute Semi-Latus Rectum.
      var r = sLR/(1 + oE * Math.cos(theta));  // Compute radial distance.
      // Compute position coordinates pos[0] is x, pos[1] is y, pos[2] is z
      pos[0] = r * (Math.cos(aP + theta) * Math.cos(aN) - Math.cos(oI) *
                    Math.sin(aP + theta)
                 *  Math.sin(aN)) ;
      pos[1] = r * (Math.cos(aP + theta) * Math.sin(aN) + Math.cos(oI) *
                    Math.sin(aP + theta)
                 *  Math.cos(aN)) ;
      pos[2] = r * (Math.sin(aP + theta) * Math.sin(oI)) ;

  return pos ;
}
```

Figure 9. Orbital propagator.

```
{
  "@context": {
    "HelioCentricOrbit": "http:// www.purl.org/ssao#Helio-Centric_Orbit",
    "CelestialEntity": "http://www.w3.org/2002/07/owl#NamedIndividual",
    "Name": "http://www.w3.org/2001/XMLSchema#Name",
    "name": "http://www.purl.org/ssao#has_name",
    "semiMajorAxis": "http://www.purl.org/ssao#has_Semi-major_Axis",
    "eccentricity": "http://www.purl.org/ssao#has_orbital_eccentricity",
    "inclination": "http://www.purl.org/ssao#has_orbital_inclination",
    "meanAnomoly": "http://www.purl.org/ssao #has_Mean_Anomaly",
    "argPerigee": "http:// www.purl.org/ssao#has_Argument_of_Periapsis",
    "sidereal": "http:// www.purl.org/ssao #has_Eccentric_Anomaly",
    "raan": "http:// www.purl.org/space-ontology/ssao#has_RAAN",
    "double": "http://www.w3.org/2001/XMLSchema#double",
    "decimal": "http://www.w3.org/2001/XMLSchema#decimal",
    "graph": "@graph",
    "type": "@type",
    "value": "@value"
  },
"graph": [ },
    { "@id": "…#Earth_Orbit",
      "type": [
        "CelestialEntity",
        "HelioCentricOrbit"
      ],
      "argPerigee": {
        "type": "decimal",
        "value": "100.47"
      },
      "meanAnomoly": {
        "type": "decimal",
        "value": "100.47"
      },
      "raan": {
        "type": "decimal",
        "value": "-11.26064"
      },
      "semiMajorAxis": {
        "type": "decimal",
        "value": "1.00000011"
      },
      "name": {
        "type": "Name",
        "value": "Earth_Orbit"
      },
      "eccentricity": {
        "type": "decimal",
        "value": "0.01671022"
      },
      "inclination": {
        "type": "decimal",
        "value": "0.00005"
      }    },…]
```

Figure 10. Part of a JSON-LD file that specifies Earth's Keplerian parameters.

## 7. CONFIGURING PROTÉGÉ FOR IMPORTING DATA

The next section presents a tutorial to import data into the SSAO using the Create Axioms from Excel workbook function. To ensure that the imported appear as named individuals with specified names, Protégé must be configured so that the instantiated object names are generated from the Excel workbook instead of autogenerated.

Click on Preferences… under the file menu to configure the identifiers assigned to objects, or entities, also known as individuals. The Preferences dialog box has several tabs as depicted in figure 11. When the dialog box appears, select the New Entities tab. Identifiers for new entities can be an auto-generated globally unique identifiers (GUIDs), a user specified name followed by sequential numbers, or a user supplied name. As depicted in figure 11, click the radio button for user supplied name. Selecting this option means that the names of the entities will come from an input data file. Next, select the Renderer tab. Click the radio button for Render by annotation property, as depicted in figure 12.

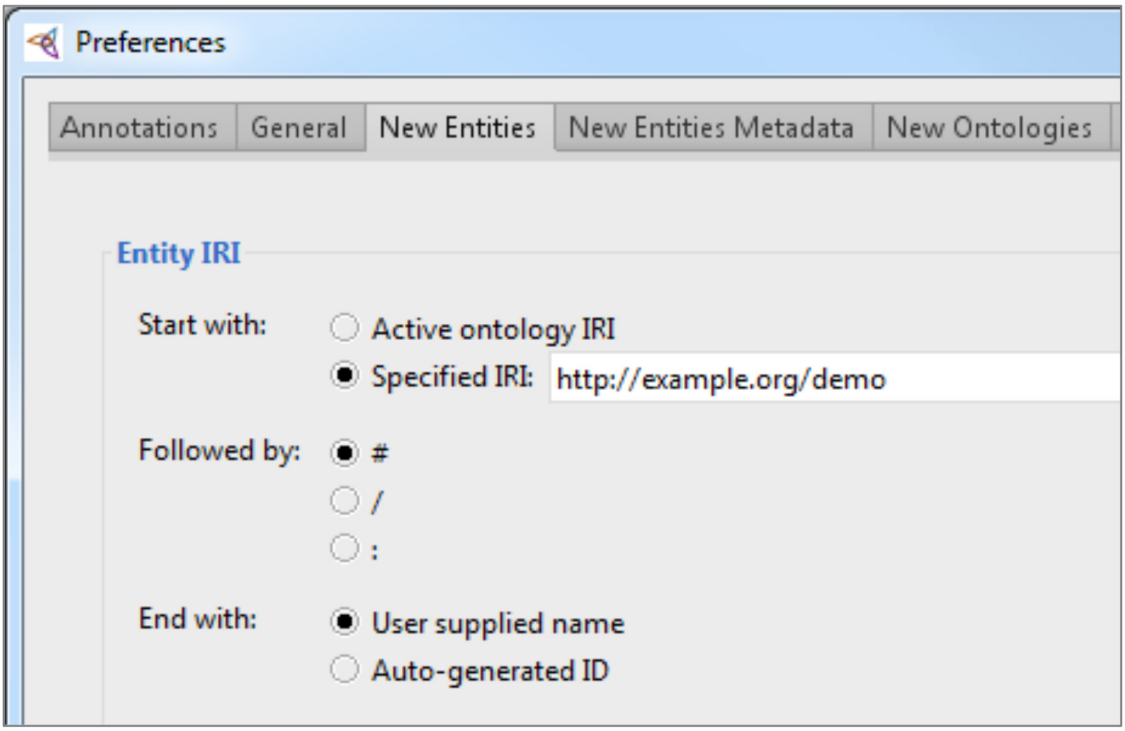



Figure 11. New Entities tab of the configure Protégé dialog box.

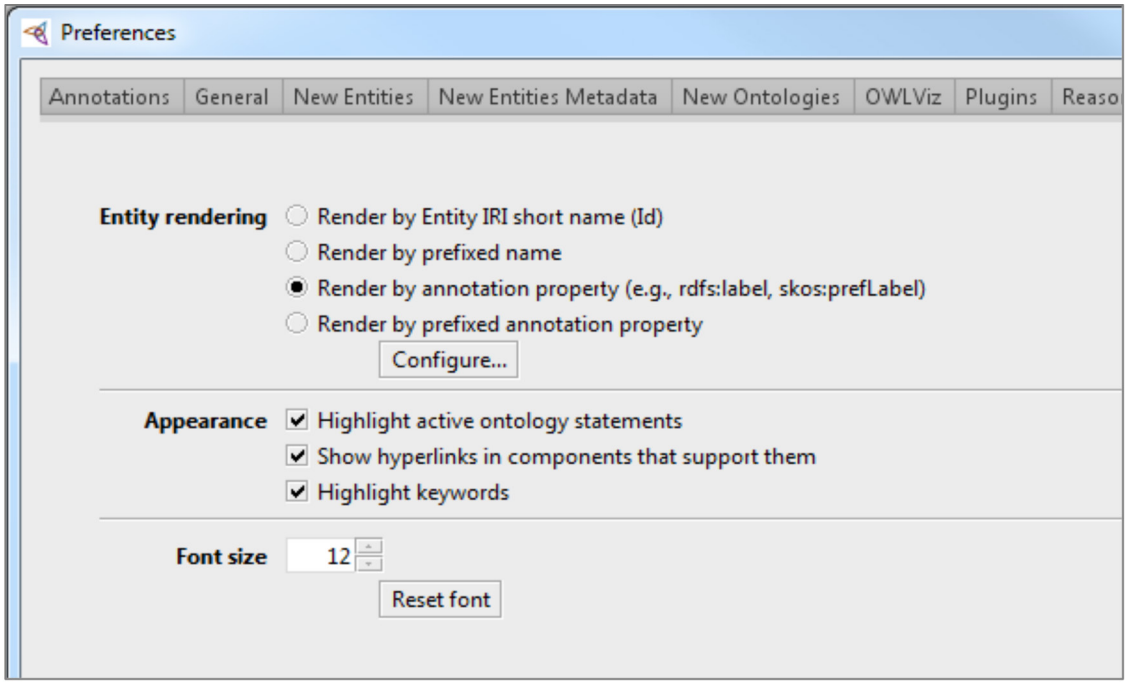



Figure 12. Renderer tab of the Protégé configuration dialog box.

Selecting this option means that an annotation property from an input data file will be used for individual identifiers. With this configuration, new entities will get their name from the imported Excel data and rendered within Protégé by those names.

## 8. TUTORIAL: ORRERY IMPLEMENTED IN R USING A JSON-LD DATA FILE

### 8.1 Overview

Several thousand code packages are available for the R programming language. A few packages provide functions for manipulating JavaScript Object Notation Linked-Data (JSON-LD) and generating HyperText Markup Language (HTML) tables and interactive 3D models. This tutorial provides a procedure for importing data into the Protégé ontology editor, exporting an expanded JSON-LD document, using an R package to convert the file format from expanded to compact, and using data from the compact JSON-LD file to generate an HTML table of planetary orbit parameters and interactive 3D plot the planetary orbits.

### 8.2 Objectives

Learning objectives for this tutorial include exporting expanded JSON-LD from Protégé, converting that file to compact JSON-LD, accessing the data via the R programming language, and generating an HTML table using the data and an interactive 3D plot from processed data. The interactive 3D plot depicts orbital trajectories of the planets; so it is a model of the solar system (known as an orrery). Skills gained from following this tutorial enable the development of ontology-based web reports.

### 8.3 Background

The tutorial about creating an expert system with the Protégé ontology editor explained how to configure Protégé for individual identifiers. As described in that tutorial, set the new entity IRIs) and Rendering settings to use the rdfs:label. These configuration settings ensure that NamedIndividuals receive the names from the imported data.

Johannes Kepler identified orbital parameters that define an elliptical trajectory of an object moving around another object. These orbital parameters[42] include:

- Semi-major axis: Half of the distance between the nearest and farthest points of the elliptical orbit. It describes the size of the orbit.

- Eccentricity: This value is a ratio of the smaller width over the larger width of the ellipse.

- Inclination: The tilt of the orbit with respect to a reference plane (e.g., equatorial plane), expressed as an angular measurement between the satellite and the reference plane.

- Longitude of ascending node: An angular measurement from a reference direction or line to the ascending node. The ascending node is a point where the orbiting object ascends through the reference plane, south to north.

- Argument of periapsis: An angular measurement from the Ascending Node to the point on the ellipse that is closest to the primary body.

- True anomaly: The current position of the secondary body (the satellite) in orbit around the primary.

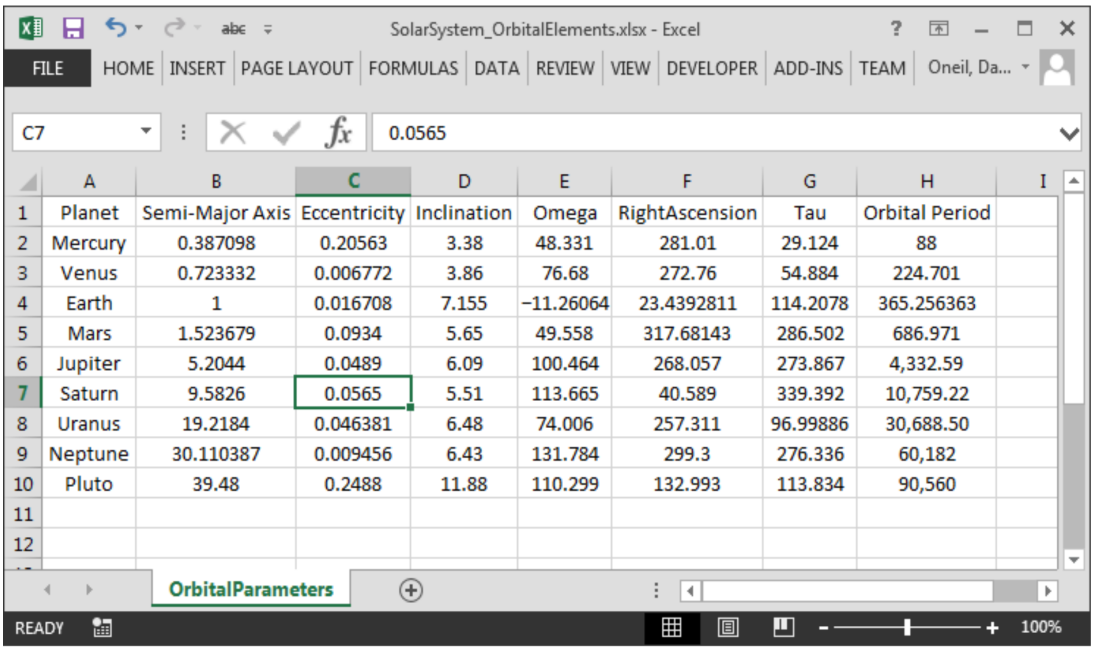
Although these parameters have corresponding ontology terms in the SSAO, the following sections present a procedure for reading an Excel spreadsheet of orbital parameters into an ontology, exporting an expanded JSON-LD document, compacting the JSON-LD document, and writing R code to generate an HTML table and plot the orbital trajectories.

### 8.4 Creating an Orbit Ontology With Protégé

After configuring Protégé as described in the background section of this tutorial, create a class named Orbit. Click on the Data Properties tab and create the following data properties or variables: eccentricity, inclination, omega, period, raan, sma, and tau. Variable inclination defines the pitch orientation of an orbit. Variable $\Omega$ will represent longitude of the ascending node, which will define the yaw orientation of the orbit. The right ascension of the ascending node (RAAN) will be represented by the variable 'raan', which defines the roll orientation of the orbit. Variable sma represents the semi-major axis (or size) of an orbit. Variable $\tau$ represents the argument of periapsis, which can serve as a starting point for simulating an orbit. The next step of the procedure involves creating an Excel workbook with orbital parameter instance data and importing the data into the orbit ontology.

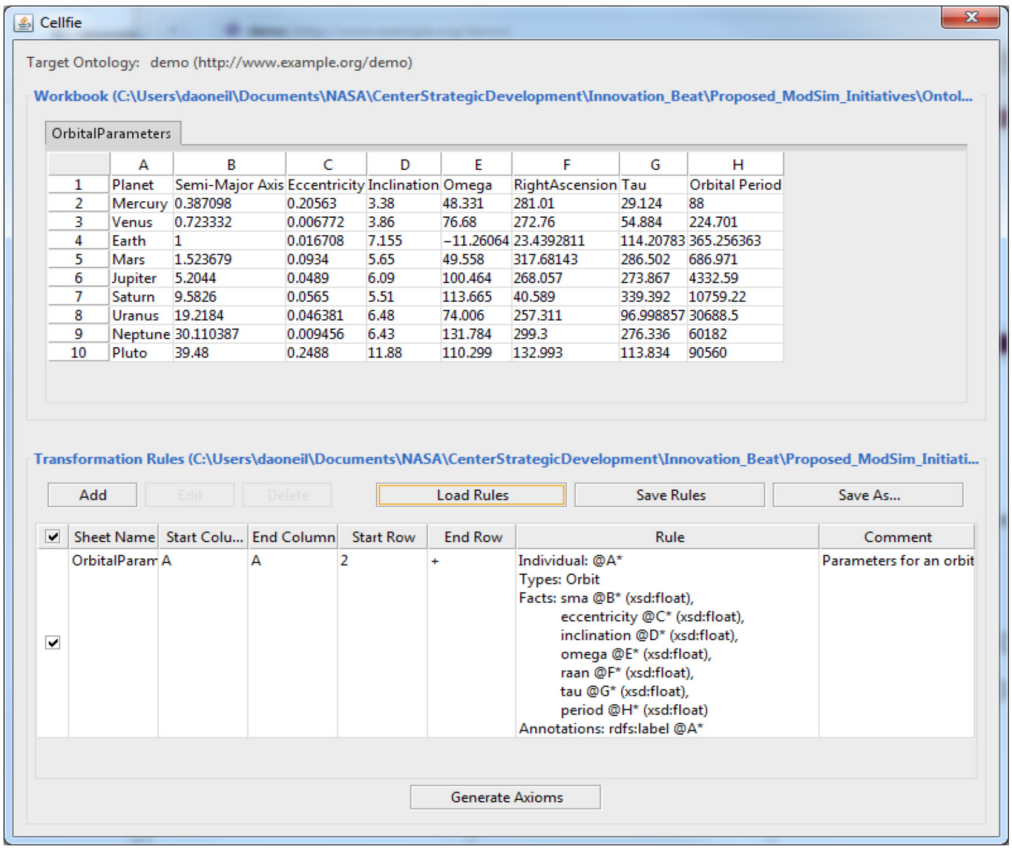
### 8.5 Importing Orbital Parameter Data Into Protégé

Create an Excel workbook with orbital parameter data. Figure 13 depicts an Excel spreadsheet with orbital parameters for the planets in our solar system. Articles in Wikipedia provide data about each planet for a spreadsheet. The tutorial about creating a simple expert system with the Protégé editor included a section that explained how to write rules to import data using the transformation rule editor. Figure 14 depicts the Cellfie dialog box, which appears upon selection of the option Create Axioms from Excel workbook option on the Tools menu in Protégé. The transformation rule editor appears upon clicking the Add button on the Cellfie dialog box. When a spreadsheet contains headers in the first row, ensure that the starting field in the transformation rule editor specifies row two, so the headers will be skipped when importing the data. In the rule, the first line, Individual: @A*, indicates creation of a NamedIndividual for each row of column A. Line two of the rule specifies the Type, which for this tutorial is the Orbit class created in the previous section.




| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Planet | Semi-Major Axis | Eccentricity | Inclination | Omega | RightAscension | Tau | Orbital Period |
| 2 | Mercury | 0.387098 | 0.20563 | 3.38 | 48.331 | 281.01 | 29.124 | 88 |
| 3 | Venus | 0.723332 | 0.006772 | 3.86 | 76.68 | 272.76 | 54.884 | 224.701 |
| 4 | Earth | 1 | 0.016708 | 7.155 | −11.26064 | 23.4392811 | 114.2078 | 365.256363 |
| 5 | Mars | 1.523679 | 0.0934 | 5.65 | 49.558 | 317.68143 | 286.502 | 686.971 |
| 6 | Jupiter | 5.2044 | 0.0489 | 6.09 | 100.464 | 268.057 | 273.867 | 4,332.59 |
| 7 | Saturn | 9.5826 | 0.0565 | 5.51 | 113.665 | 40.589 | 339.392 | 10,759.22 |
| 8 | Uranus | 19.2184 | 0.046381 | 6.48 | 74.006 | 257.311 | 96.99886 | 30,688.50 |
| 9 | Neptune | 30.110387 | 0.009456 | 6.43 | 131.784 | 299.3 | 276.336 | 60,182 |
| 10 | Pluto | 39.48 | 0.2488 | 11.88 | 110.299 | 132.993 | 113.834 | 90,560 |

Figure 13. Excel spreadsheet with planetary orbit parameters.




| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Planet | Semi-Major Axis | Eccentricity | Inclination | Omega | RightAscension | Tau | Orbital Period |
| 2 | Mercury | 0.387098 | 0.20563 | 3.38 | 48.331 | 281.01 | 29.124 | 88 |
| 3 | Venus | 0.723332 | 0.006772 | 3.86 | 76.68 | 272.76 | 54.884 | 224.701 |
| 4 | Earth | 1 | 0.016708 | 7.155 | −11.26064 | 23.4392811 | 114.20783 | 365.256363 |
| 5 | Mars | 1.523679 | 0.0934 | 5.65 | 49.558 | 317.68143 | 286.502 | 686.971 |
| 6 | Jupiter | 5.2044 | 0.0489 | 6.09 | 100.464 | 268.057 | 273.867 | 4332.59 |
| 7 | Saturn | 9.5826 | 0.0565 | 5.51 | 113.665 | 40.589 | 339.392 | 10759.22 |
| 8 | Uranus | 19.2184 | 0.046381 | 6.48 | 74.006 | 257.311 | 96.998857 | 30688.5 |
| 9 | Neptune | 30.110387 | 0.009456 | 6.43 | 131.784 | 299.3 | 276.336 | 60182 |
| 10 | Pluto | 39.48 | 0.2488 | 11.88 | 110.299 | 132.993 | 113.834 | 90560 |

Figure 14. Dialog box for the Create Axioms from Excel workbook tool option.

The Facts section of the rule maps the columns to the Data Properties and specifies a data type. The last line of the rule specifies column A as containing the label for the NamedIndividuals. If the renderer is configured to use labels, the Named Individuals will have the labels as their names. Figure 15 depicts the "Individuals by Class" tabbed window after importing the orbital parameter data. The next section explains how to compact an expanded JSON-LD file via R code.

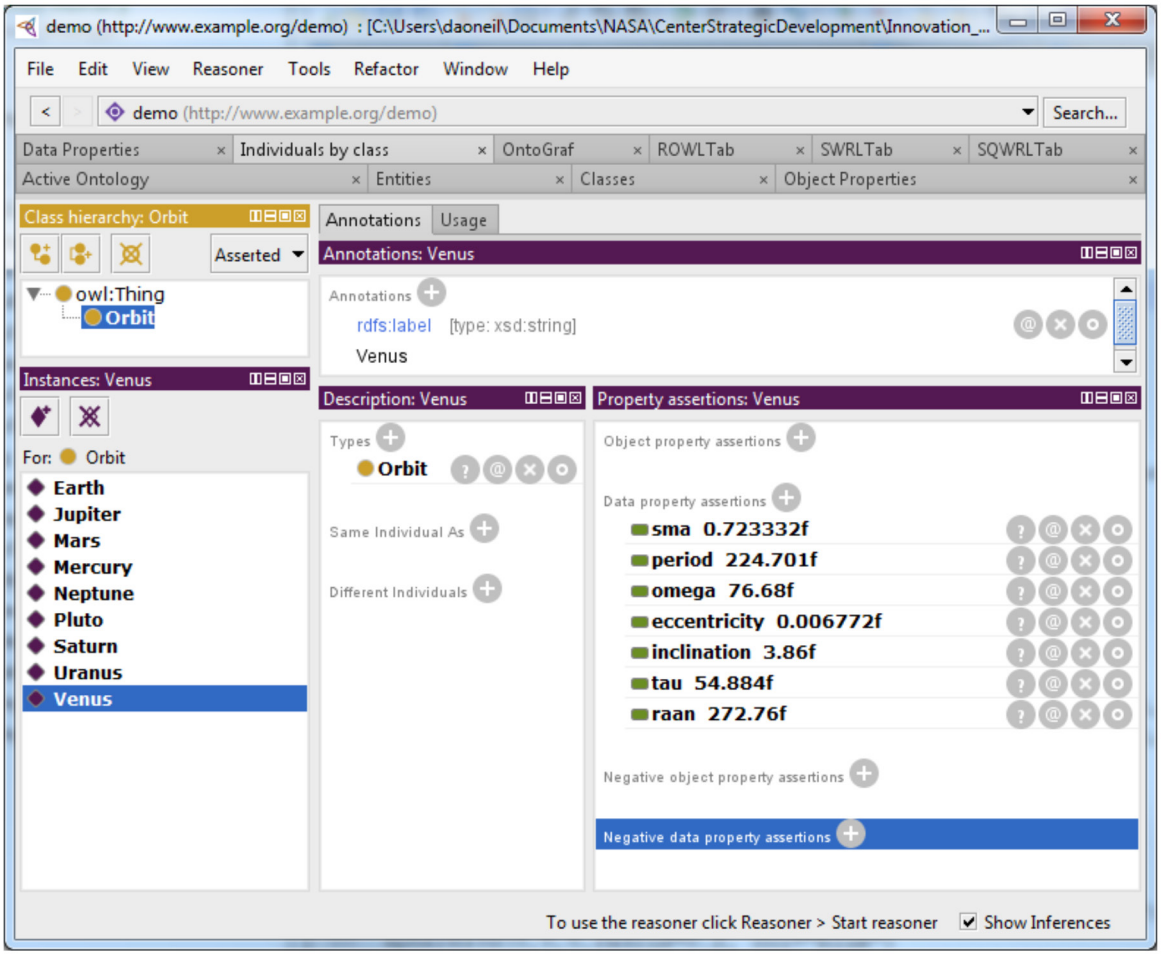



Figure 15. Named individuals with populated data properties after importing the data.

### 8.6 Compacting an Expanded JSON-LD Document

Protégé can save a file in the expanded JSON-LD format. Under the File menu, select the option to Save the file.[43] When the Save dialog box appears, click the drop-down list button and select JSON-LD. Figure 16 presents a few lines from a JSON-LD file saved from Protégé. Notice that each @type is followed by an explicit IRI. During compaction, these IRIs will be replaced with a brief term. Another detail to notice is the value for omega. This value for Earth is supposed to be a negative number; evidently, there exists a problem with negative values in the JSON-LD export. In the next section, some R code will fix this problem.

```
{
  "@graph": [
    {
      "@id": "http://example.org/demo#Earth",
      "@type": [
        "http://example.org/demo#Orbit",
        "http://www.w3.org/2002/07/owl#NamedIndividual"
      ],
      "http://example.org/demo#eccentricity": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "0.016708"
      },
      "http://example.org/demo#inclination": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "7.155"
      },
      "http://example.org/demo#omega": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "\\u221211.26064"
      },
      "http://example.org/demo#period": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "365.25638"
      },
      "http://example.org/demo#raan": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "23.439281"
      },
      "http://example.org/demo#sma": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "1.0"
      },
      "http://example.org/demo#tau": {
        "@type": "http://www.w3.org/2001/XMLSchema#float",
        "@value": "114.20783"
      },
      "http://www.w3.org/2000/01/rdf-schema#label": "Earth"
    },….
```


Figure 16. Part of an expanded JSON-LD file exported from Protégé.

The JSON-LD R package documentation provides an example for the jsonld_compact function.[44] Input values to the jsonld_compact function include an expanded JSON-LD document and a variable containing a character string that specifies a context that defines terms to substitute

for each IRI. Figure 17 presents content for a context variable. Notice that when a term is defined in a context it can be used later in the context, e.g., notice how ‘demo’ and ‘float’ appear in term definitions. Another detail to notice in this snippet is the defined term ‘graph,’ which will replace the @graph in the compact JSON-LD format. Eliminating the @ symbol will enable the R code to access the data.

```
{
    "demo": "http://example.org/demo#",
    "float": "http://www.w3.org/2001/XMLSchema#float",
    "owl": "http://www.w3.org/2002/07/owl#",
    "label": "http://www.w3.org/2000/01/rdf-schema#label",
    "comment": "http://www.w3.org/2000/01/rdf-schema#comment",
    "eccentricity": {
      "@id": "demo:eccentricity",
      "@type": "float"
    },
    "inclination": {
      "@id": "demo:inclination",
      "@type": "float"
    },
    "omega": {
      "@id": "demo:omega",
      "@type": "float"
    },
       "omega": {
      "@id": "demo:omega",
      "@type": "float"
    },
    "sma": {
      "@id": "demo:sma",
      "@type": "float"
    },
    "raan": {
      "@id": "demo:raan",
      "@type": "float"
    },
    "tau": {
      "@id": "demo:tau",
      "@type": "float"
    },
    "period": {
      "@id": "demo:period",
      "@type": "float"
    },
    "graph": "@graph"
  }
```

Figure 17. Compact version of the JSON-LD file.

Figure 18 presents R code to load the JSON-LD, jsonlite,[45] and htmltable libraries. Before these libraries can be loaded, the packages must be installed. The command install.packages will open the R documentation about installing packages. After the libraries are loaded the line fname <- "justContext.txt" provides the name of the text file containing the content presented in figure 16. Notice there is no pathname, which means that the file exists within the working directory used by the R interpreter. The R command, setwd, changes the working directory to a specified folder. The next line in figure 18 reads the contents of the specified file as a character string. The second input parameter of the readChar command gets the size of the file to determine the length of the character string. With the content from figure 16 stored in the variable named orbContext, the R code in figure 18 can call the jsonld_compact function to apply the context to compact the expanded JSON-LD document.

```
install.packages("jsonlite","jsonld","htmlTable")
library(jsonlite)
library(jsonld)
library(htmlTable)

fname <- "justContext.txt"
orbContext <- readChar(fname,file.info(fname)$size)

compactOrbits <- jsonld_compact("PlanetaryOrbits.owl",orbContext)
```

Figure 18. R code that reads a context character string and compacts the JSON-LD file.

The JSON-LD Playground web application enables the development of a context section to compact an expanded JSON-LD file. If you use the JSON-LD Playground to compact the file, the last three lines of code in figure 18 can be skipped and the fromJSON function from the jsonlite library can read the compact JSON-LD file into a variable. Figure 19 presents a screenshot of the JSON-LD Playground with the new JSON-LD Context field for defining terms to replace the IRI. Text in the new JSON-LD Context field is the same as the text in figure 10. A first line { "@context:" } has been added along with a corresponding closing curly bracket. As the context develops, one can see the IRI being replaced in the compacted frame of the JSON-LD Playground webpage. With a compact JSON-LD document, R code can read and process the data. The next two sections present code that generates an HTML table and a 3D plot.

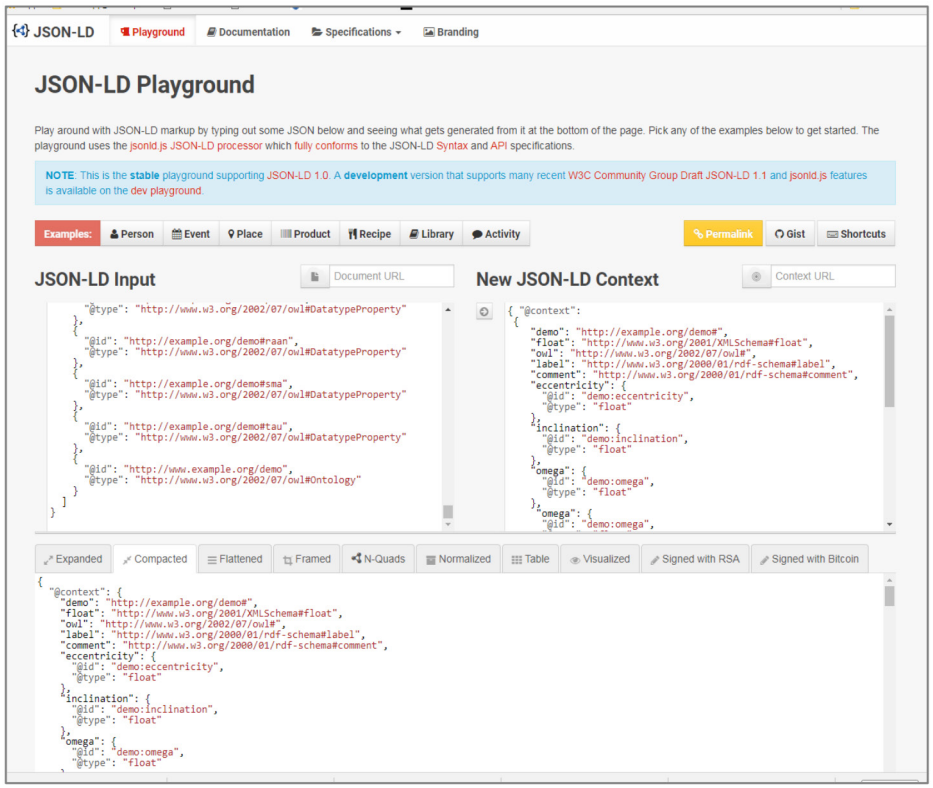



Figure 19. The JSON-LD Playground with the New JSON-LD Context field.

### 8.7 Generating an HTML Table

With the jsonlite and htmltable packages installed, it only takes a few lines to produce an HTML table from a compact JSON-LD document. Figure 20 presents R code that uses the fromJSON function to load the compact JSON-LD into a variable named Orbits. Recall that the last line of the code in figure 17 defined the term 'graph' to eliminate the @ symbol. The second line of the code in figure 20 extracts columns 3–10 from orbits$graph and stores the matrix in a variable named orbitparms. The na.omit function removes the Not Available (NA) data from the matrix. Recall the problem with the negative number in figure 16; a line of code in figure 20 replaces the value in the matrix to fix that problem. Notice the comment about determining the coordinates. Typing the name of the variable orbitparms at the command prompt lists the table so you can determine the coordinates of the number that needs to be replaced. Values in the orbitparms matrix are characters, so an input parameter in the order function converts the characters to numbers via the as.numeric function. The specified column for the ordering is the semi-major axis (sma). The htmlTable function formats the matrix as HTML. The next to last line establishes a file connector and the writeLines function writes the HTML to a file. Table 1 was copied from the generated HTML file as it was rendered in a web browser and pasted into this document.

```
orbits <- fromJSON(compactOrbits,flatten=TRUE)
orbitparms <- orbits$graph[,3:10]
orbitparms <- na.omit(orbitparms)
# Ensure that the columns are in the correct order.
orbitparms <- orbitparms[c("label","eccentricity","inclination",
                           "omega","period","raan","sma","tau")]
orbitparms                    # print the variable to the screen
orbitparms[1,4] <- -11.26064  # determine the coordinates (it may not be 1,4)
orbitparms <- orbitparms[order(as.numeric(orbitparms$sma)),]
orbparmTable <- htmlTable(orbitparms,rnames=FALSE)
fcon <- file("orbitingplanets.html")
writeLines(orbparmtable,fcon)
```

Figure 20. R code that generates an HTML table.

Table 1. HTML table generated from the R code.

| Label | Eccentricity | Inclination | Omega | Period | RAAN | SMA | Tau |
|---|---|---|---|---|---|---|---|
| Mercury | 0.20563 | 3.38 | 48.331 | 88.0 | 281.01 | 0.387098 | 29.124 |
| Venus | 0.006772 | 3.86 | 76.68 | 224.701 | 272.76 | 0.723332 | 54.884 |
| Earth | 0.016708 | 7.155 | –11.26064 | 365.25638 | 23.439281 | 1.0 | 114.20783 |
| Mars | 0.0934 | 5.65 | 49.558 | 686.971 | 317.68143 | 1.523679 | 286.502 |
| Jupiter | 0.0489 | 6.09 | 100.464 | 4332.59 | 268.057 | 5.2044 | 273.867 |
| Saturn | 0.0565 | 5.51 | 113.665 | 10759.22 | 40.589 | 9.5826 | 339.392 |
| Uranus | 0.046381 | 6.48 | 74.006 | 30688.5 | 257.311 | 19.2184 | 96.998856 |
| Neptune | 0.009456 | 6.43 | 131.784 | 60182.0 | 299.3 | 30.110388 | 276.336 |
| Pluto | 0.2488 | 11.88 | 110.299 | 90560.0 | 132.993 | 39.48 | 113.834 |

### 8.8 Generating an Interactive 3D Plot of the Planetary Orbits

With the orbital parameters stored in an *R* variable, additional code can generate sets of 3D coordinates and plot lines that represent the planetary orbital trajectories. Code presented in this section generate sets of 3D coordinates for an ellipse and use functions from the rgl R package plot the orbital trajectories. Keplerian parameters are used in the generation and rotation of ellipses that represent orbits.

Figure 21 presents functions for generating coordinates for an ellipse and rotating the ellipse coordinates. Input variables for the function makeOrbit include the semi-major axis (sma), eccentricity, inclination, omega, and raan; the function generates 80 points and returns a set of 3D coordinates. The positSphere function generates 80 points between negative and positive $\pi$. The $\tau$ column in table 1 represents a starting position for the planets within their orbital trajectories. After converting tau to radians and adjusting it to be in the same range as the theta values, the positSphere function finds the closest value of theta to tau and gets the index of the array. That

index, trueAnomaly, is used to set the 3D coordinates for the position of a sphere. Inputs to the positSphere function include the size and color for the planet. The planetaryOrbit function uses the Keplerian parameters from the orbit arms table and a planet number to get the data, convert it from characters to numbers, and call the make Orbit function.

```
positSphere <- function (planet, orbitnum, size, color) {
   # Produce a sequence of potential true anomaly values.
     theta <- seq(-pi,pi, length.out=80)
     tau <- as.numeric(orbitparms[orbitnum,8]) * (pi/180)
       tau <- tau - pi        # adjust tau to be in range of theta

  # Find the theta that is closest to tau.
     diffThetaTau <- abs(theta) - abs(tau)
     trueAnomoly <- which(diffThetaTau == min(diffThetaTau))
     taX <- planet$coordinates$x[trueAnomoly]
     taY <- planet$coordinates$y[trueAnomoly]
     taZ <- planet$coordinates$z[trueAnomoly]
     sphereid <- spheres3d(xyz.coords(x=taX,y=taY,z=taZ),
                           radius = size, col = color)
}

planetaryOrbit <- function(kp,n) { # Keplerian parameters and planet number
   a <- as.numeric(kp[n,7])   # semi-major axis
   e <- as.numeric(kp[n,2])   # eccentricity
   oI <- as.numeric(kp[n,3])  # orbital inclination
   aN <- as.numeric(kp[n,6])  # right ascension of ascending node
   aP <- as.numeric(kp[n,4])  # argument of perihelion

   helioOrbit <- makeOrbit(a,e,oI,aN,aP)
   return(helioOrbit)
}
```

Figure 21. R code for generating and rotating an ellipse based on orbital parameters.

Code in figure 22 opens a 3D plotting device with the decorate3d function from the rgl library[46] that was loaded by the code at the top of figure 23. Parameters of the decorate3d function turn off labels, a bounding box, and axes. The next two lines of the code in figure 22 establish color and size arrays for the spheres that will represent the planets. A list object receives a generated planetary trajectory comprised of 3D coordinates. Subsequently, a 'for loop' appends the other planetary orbital trajectories. A second 'for loop' increments an index to plot the planetary orbital trajectories and the colorful spheres that represent the planets. Loading the htmltools library enables use of the save_html function at the end of the code snippet. The scene3d and rglwidget functions, from the rgl library store the plot as a scene and converts that scene to a widget. The save_html function generates an HTML page that presents an interactive 3D plot. The next section explains how to integrate the table and the 3D plot into a single webpage.

```
# Open a 3D plotting device that has no labels, box, or axes.
decorate3d(xlab="", ylab="",zlab="",box=FALSE, axes=FALSE)
# Establish arrays of colors and sizes for the planets.
orbcolors <-c("brown","gray","blue","red","orange","darkorchid",
               "cadetblue","aquamarine4","blue4")
orbsizes <- c(0.015,0.04,0.04,0.03,1,0.8,0.7,0.6,0.02)

planets <- planetaryOrbit(orbitparms,1)  # Generate trajectory of planet 1.

for(i in 2:9) {   # Generate the other trajectories.
    planets <- c(planets,planetaryOrbit(orbitparms,i))
}
for(i in 1:9) {   # Plot the trajectories and position the planets.
  plot3d(planets[i]$coordinates,add=TRUE,type="l",col=orbcolors[i])
  positSphere(planets[i],i,orbsizes[i],orbcolors[i])

}
library(htmltools)             # load htmltools for the save_html function
orrery <- scene3d()            # Store the plot as a scene.
orbwidget <- rglwidget(orrery) # Convert the scene to an rgl widget.
save_html(orbwidget,"orrery.html") # Save the widget as a web page.
```

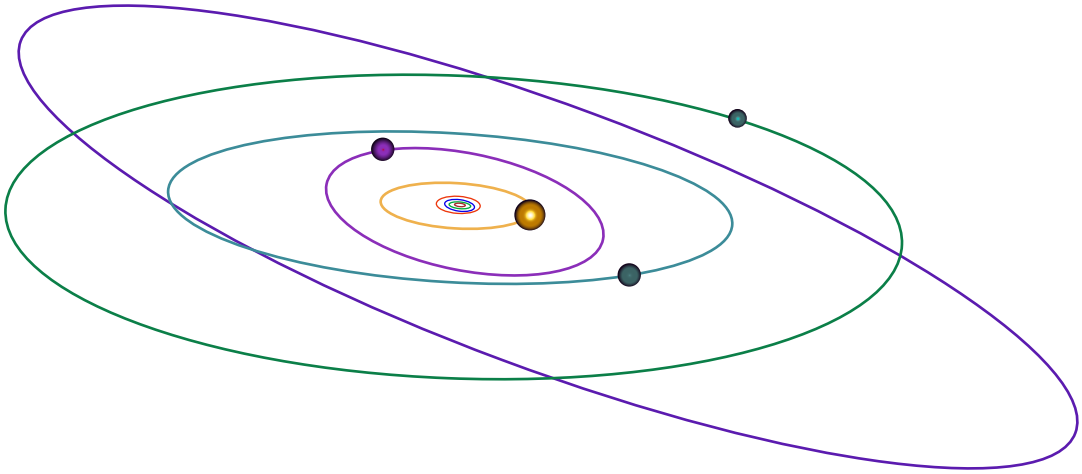
Figure 22. R code to generate a 3D plot of the elliptical orbits.

```
  install.packages("rgl")
  library(rgl)      # Load the rgl library for 3D rotations and plotting.

  makeOrbit <- function(a,e,oI,aN,aP) {
      oI <- oI * 0.01745329 # convert orbital inclination to radians
      aN <- aN * 0.01745329 # convert longitude of ascending node to radians
      aP <- aP * 0.01745329 # convert argument of perihelion to radians
      sLR = a * (1 - e^2)   # Compute Semi-Latus Rectum.

     theta <- seq(-pi,pi, length.out=80)
     r = sLR/(1 + e * cos(theta))    # Compute radial distance.
   x <- r * (cos(aP + theta) * cos(aN) - cos(oI) * sin(aP + theta) * sin(aN))
   y <- r * (cos(aP + theta) * sin(aN) + cos(oI) * sin(aP + theta) * cos(aN))
   z <- r * (sin(aP + theta) * sin(oI))

  points <- xyz.coords(x,y,z)
  helioorbit <- list(coordinates = points )
  return(helioorbit)
}
```

Figure 23. R code to convert orbital parameters to Cartesian coordinates.

### 8.9 Integrating the HTML Files With a Text Editor

Figure 20 presented an R script that generated an HTML table. Figure 22 presented an R script that generated an HTML file with embedded scripts that call library functions to generate an interactive 3D model of the solar system. Integrating the two HTML files is simply a matter of opening both of them and copying and pasting the <table>….</table> HTML code into the body of the other HTML file between the <script> and </body> tags. Figure 24 presents the integrated webpage with the interactive 3D model and HTML table.



| Label | Eccentricity | Inclination | Omega | Period | Raan | SMA | Tau |
|---|---|---|---|---|---|---|---|
| Mercury | 0.20563 | 3.338 | 48.331 | 88.0 | 281.01 | 0.387098 | 29.124 |
| Venus | 0.006772 | 3.86 | 76.68 | 224.701 | 272.76 | 0.723332 | 54.884 |
| Earth | 0.016708 | 7.155 | –11.26064 | 365.25638 | 23.439281 | 1.0 | 114.20783 |
| Mars | 0.0934 | 5.65 | 49.558 | 686.971 | 317.68143 | 1.523679 | 286.502 |
| Jupiter | 0.0489 | 6.09 | 100.464 | 4332.59 | 268.057 | 5.2044 | 273.867 |
| Saturn | 0.0565 | 5.51 | 113.665 | 10759.22 | 40.589 | 9.5826 | 339.392 |
| Uranus | 0.046381 | 6.48 | 74.006 | 30688.5 | 257.311 | 19.2184 | 96.998856 |
| Neptune | 0.009456 | 6.43 | 131.784 | 60182.0 | 299.3 | 30.110388 | 276.336 |
| Pluto | 0.2488 | 11.88 | 110.229 | 90560.0 | 132.993 | 39.48 | 113.834 |

Figure 24. Integrated HTML table and interactive 3D model.

### 8.10 Tutorial Summary

This tutorial explained a procedure for importing data from Excel into Protégé, generating an expanded JSON-LD document from Protégé, provided two methods for compacting the JSON-LD file, and demonstrated how to generate an HTML table and an interactive 3D plot. With a capability to read JSON-LD and post process the data into webpages with tables and interactive 3D plots, an ontology can manage mission data.

## 9. A VISION FOR AN ONTOLOGY-BASED ORRERY

A free, open-source, web-based solar system model, or orrery, could serve as a graphical user interface to an ontology generated database. This section presents a vision for a notional ontology based orrery. Information about celestial bodies and spacecraft could pop-up as a user interacts with the 3D model. Code could use Application Programming Interfaces (APIs) to request data to be presented via the web-based orrery. Potential use cases include education, conveying concepts, presenting an integrated view of future space industries, and enabling the development of serious games.

### 9.1 Education

Educators and students could explore the solar system through the orrery part of the orrery. A user interface with time line and dial for accelerating time could provide the capability to view celestial events such as solar eclipses, visits from Halley's comet, or the Tunguska event. Other user interface widgets could provide a capability to measure distances between various celestial bodies and estimate travel times to get from point A to point B at various velocities.

### 9.2 Conveying Mission Concepts

Mission planners could use the orrery to identify way points and orbital maneuvers. A 3D model repository could enable planners to either select existing spacecraft models or upload a new model. By assigning velocities to the mission plan and selecting a cockpit view, mission planners could simulate the mission and see the Solar System from the perspective of the spacecraft as it moves among the way points. Mission planners could save the way points and make them public or limit access to a list of people by created an e-mail distribution list. Mission plans saved for public viewing can educate the general public. A feature of the mission plans could include rationale for maneuvers and way points to provide insight and a basis for discussion. Space agencies could incorporate interactive mission visualizations into their websites.

### 9.3 Integrated View of Emerging Space Industries

Space entrepreneurs could use the orrery to augment their business plans by depicting service routes, traffic models, and service expansion. A digital globe would be better suited for depicting commercial activities in Low Earth Orbit, such as satellite servicing, orbital debris collection, or orbiting hotels. A orrery would better serve depiction of commercial space travel to the Moon, asteroid mining, or commercial transportation among space colonies. Space entrepreneurs could use the 3D model repository to deploy spacecraft and the missing planning tools to depict anything from assaying asteroids to deploying space infrastructure for communications, power, and manufacturing. Corporations could integrate interactive space operations models into their websites.

## 9.4 Requirements for an Ontology-Oriented Orrery

The Use Cases identified a few of the proposed orrery functions including mission planning tools, various camera views, a time line, a time acceleration widget, and a 3D model repository. Proposed features include native execution with a modern web browser, an API that enables the development of web-based applications, and an open source development approach that enable contributions from everyone. The following subsections provide details about these functions and features.

### 9.4.1 Native Execution within a Web Browser

The capability to use the orrery without the requirement to download a thick-client, install a plug-in, or update a virtual machine will make the system more attractive to the casual user. Programming language such as JavaScript and WebGL execute natively within popular web browsers and take advantage of graphics processors.

### 9.4.2. Planets, Moons, and Asteroids

An ideal web-based orrery would include all of the known planets, moons, and asteroids; however, rendering detailed models of the 600,000 known asteroids and objects within the rings of Saturn could overwhelm a web browser. Menu items or check boxes could provide a capability to configure orrery so that it presents the most interesting celestial bodies to the individual. One website visitor may want to view Earth and a selected set of potentially hazardous asteroids (PHA). Another visitor may want to view Jupiter and its moons.

### 9.4.3 Level of Detail

Managing level of detail can reduce the load on the graphics processors. From a distance, the asteroid belt could appear as particles and the planets could be rendered with procedural textures. As the point of view moves closer to a planetary surface, a texture library could provide a more technically accurate texture map. Asteroids that have shape files could have corresponding 3D models. From a distance, a generic ovoid shape could represent the asteroid. As a camera moves closer to the asteroid, a more accurate 3D model could be substituted for the generic model.

### 9.4.4 Time and Navigation Widgets

User interface widgets for managing time and selecting points of view will prevent the visitor from getting lost in the orrery. An initial set of widgets include:

- Time line: A configurable timeline could present minutes for an approach to an asteroid, weeks for the depiction of a spacecraft conducting an orbital maneuver, months for a mission to Mars, years to depict comet and meteor trajectories, and centuries or millennia to show long term behaviors of the Solar System.

- Time accelerator: A widget, such as a dial or slider can provide a visitor with a capability to accelerate a mission visualization. Depending on the implementation orbital propagator(s), time acceleration could introduce errors.

- Way Point Selection: Another method for time acceleration could be a widget or menu for selecting previously defined way points. When a visitor selects a way point, the system could reset the orbital propagator(s) to mitigate the accrual of errors.

- Camera and view selection: A function for establishing and pointing cameras could enable the presentation of a mission concept from several perspectives. When a visitor selects a spacecraft model, the system could establish a camera at the bough of the ship to depict the spacecraft's point of view. If the mission involves multiple spacecraft, each ship could have a selectable camera.

### 9.4.5 Mission Planning

The Use Cases section describe how mission planners and space entrepreneurs could use the mission planning functions to establish way points to present a mission plan. Mission planning widget and screens include:

- Map Views: Working in 3D can be difficult because the two-dimensional computer monitor is presenting a three-dimensional volume. Selectable 2D map views can simplify the process of placing way points. A visitor could select two map views and place the point on those two views to generate the 3D location. The map views ought to include selectable center points and distance scales. A visitor could start with Earth as a center point and after establishing a few points, he or she could select Mars as the center point to establish additional way points.

- Way Point Definition: A way point identifies a location in space where a spacecraft will start, travel through, or complete a mission. Way points could be represented with a variety of user selectable colored 3D icons. When creating a way point, a visitor could enter paragraphs to explain the rationale for placement and links to supporting information.

- Trajectory Trace Path: As a mission simulation executes, a spacecraft moves along a defined path through a set of way points. A detailed analysis of orbital trajectories could be done beforehand with design and analysis tools like NASA's General Mission Analysis Tool (GMAT) and Analytical Graphics Incorporated's System Tool Kit (STK). The orrery could offer a simple orbital propagator to move a spacecraft along a trajectory, provided that the visitor supplies the orbital elements for each leg of a mission plan. If the visitor does not supply the orbital elements, then the WMVS could offer straight lines or Bezier curves between way points. During the simulation, the WMVS could trace the trajectory paths with visitor specified colors, thicknesses, and line types.

### 9.4.6 Model Library

A model library could include 3D geometric meshes of asteroids, spacecraft, and other objects. Features of the model library include a 2D image, description, and dimensions of the models in the library. Import Articulated 3D models—a spacecraft model may need to reconfigure itself during a mission; for example, it could deploy solar panels and antenna. Computer modeling programs, such as Blender, provide a capability to create models with rigs and articulated movements. The import function ought to support a file format that includes the rig and articulations. The mission planning way point editor could include an option to activate the model's articulated behavior. Import Texture Maps—a file format of the 3D model ought to include texture maps so that the mission planner does not have to manipulate the texture maps within the orrery. There exists a variety of powerful free 3D modeling programs, such as Blender, which can export a COLLADA file that includes texture maps.

### 9.4.7 Application Programming Interface

The public orrery ought to provide a core set of functions and features with an API that enables developers to extend system capabilities and to create new applications that use the orrery code base. The following subsections describe interfaces to external databases with data about celestial bodies, usage of existing computer graphics code libraries, and access to the Solar System scene-graph.

### 9.4.8 Interfaces to Celestial Body Databases

The Jet Propulsion Laboratory Near Earth Objects (NEO) Project Office provides access to a database of orbital trajectories for PHAs. The Asterank website provides Representational State Transfer (REST) protocol interface to its database. The orrery API ought to provide functions for calling these external databases and other sources of celestial body data. Considering that accessing these external sources during a simulation could slow down the system, the orrery may need a local database that is periodically updated as JPL and other organizations update orbital element data. These external database data request functions could provide developers the capability to get the latest data during an initialization phase.

### 9.4.9 Interfaces to JavaScript and WebGL Libraries

Instead of creating a new 3D graphics code library, the orrery ought to build upon an existing code library, such as X3Dom, ThreeJS, or BabylonJS. If the development team can identify multiple compatible 3D graphics, data visualization, and user interface code libraries, these API functions could provide a capability to program at a higher level. Consider the mission planning tools: There could be calls to three or four different code libraries, an orrery API function that enables the developer to work in an object-oriented fashion to establish map views, way point editors, and trajectory trace paths.

### 9.4.10 Solar System Scene Graph

A scene-graph is a data structure that defines an integrated collection of 3D objects. A Solar System scene-graph at the heart of the orrery would include the planets, moons, and asteroids. A user interface would configure the scene-graph to add or remove objects. Functions in the orrery API could enable a code developer to navigate, read, or modify the scene-graph for the purpose of adding data or updating data visualization overlays.

### 9.4.11 Celestial Object Builder Utility

A future set of functions could provide a capability to build simple planetoid models. Presently, only a few shape files of asteroids exist. To provide a more sophisticated model of an asteroid than the simple ovoid shape, the celestial object builder utility functions could enable a visitor code developer to import 3D models and specify landing spots or locations of desirable resources.

## 10. SPACE SITUATIONAL AWARENESS ONTOLOGY DRAFT SPECIFICATION

Revision: 0.5
Author: Robert John Rovetto
http://orcid.org/0000-0003-3835-7817
ontologos@yahoo.com, rrovetto@terpalum.umd.edu
Imported Ontologies: Dublin Core terms

Related Publications:
An Ontological Architecture for Orbital Debris Data, online Link 1, (PDF) (2015)[10]
Preliminaries of a Space Situational Awareness Ontology, online Link 1, (PDF) (2016)[8]
An Ontology for Satellite Databases, (PDF) (2017)[11]
Orbital Debris Ontology, Terminology, and Knowledge Modeling Link 1, (PDF) (2019)[13]
Project Webpages:
https://purl.org/space-ontology
<https://purl.org/space-ontology/ssao>[12]
<https://ontospace.wordpress.com/>[11]
GitHub Webpages:
https://github.com/rrovetto/[30]
<http://rrovetto.github.io/Orbital-Space-Ontology-Project/>[47]
<http://rrovetto.github.io/space-situational-awareness-domain-ontology/>[36]
Download serialization: JSON-LD, RDF/XML, OWL, N Triples, Turtle.

### 10.1 Abstract

The Space Situational Awareness ontology (SSAO) is an ontology for describing the concepts, data, knowledge, objects and events of interest in space situational awareness (SSA) for astronautics and astronomy. It focuses on the orbital space environment, orbital debris, and spacecraft. It aims to support SSA and safe spaceflight. It provides common terminology, metadata, and knowledge model for the community. The SSAO is a work in progress, subject to change, and open to contributors and partnerships. The following subsections present a partial documentation of a provisional OWL version. Contact the author about more complete versions.

### 10.2 Disclaimer

The SSAO is a living work, and all content, including title, scope, ontology elements/constructs (classes, properties, terms, etc.), structure, definitions, etc., is subject to revision as development continues. The Ontology Based Virtual Orrery used and successfully applied, in 2016–2017, an early OWL file version of the ontology. No claims to completeness are made. The ontology is presented 'as is.' Any warranties are disclaimed, and in no event shall the copyright holders or contributors be liable for any damages arising in any way out of use of it.

## 10.3 Description

The Space Situational Awareness ontology (SSAO) is an ontology describing the concepts, data, knowledge, objects and events of interest in space situational awareness (SSA) for astronautics and astronomy. It is one ontology of a set of ontologies for astronautics.[6,10,13]

It aims to formally represent objects and activities in the orbital space environment surrounding a celestial body such as Earth, but also the infrastructure (both technological and social) required to monitor that environment. It models our knowledge of these entities, the processes by which we attain that knowledge, together with requisite data and infrastructure. The ontology defines categories, relationships, and instances to capture generic and specific domain knowledge of the target universe of discourse—the SSA domain (sometimes called 'space domain'). The categories in the ontology can serve as metadata, semantically annotating data about objects in orbit. Defined terms can serve as a standard vocabulary for the community.

SSA involves past, present, and predictive knowledge about the orbital and near space environment and the processes through which we attain that knowledge and associated data. For example, objects of interest include spacecraft orbiting Earth and datasets about their orbits and motion. Processes of interest include observation, detection, tracking and identification, as well as the prediction (and simulation) of spacecraft future trajectories. This involves collecting, processing, analyzing and disseminating data from various sources, such as ground-based sensors. The SSA domain is, therefore, tentatively construed to include the orbital space and near-space environments, occurrences therein, cataloged data about these entities ,and associated general knowledge (e.g., scientific laws or principles of astrodynamics). Purposes and goals of the ontology(s) include:

- To support safe spaceflight and space situational awareness.
- To provide metadata to link, search, and semantically annotate data and information systems.
- To provide a semantic layer for space object catalogs, databases, and information systems.
- To accurately represent objects in orbit, their orbits; observation data, processes, space actors, and infrastructures; etc.
- To provide a neutral, accurate, widely-applicable, common vocabulary, set of reusable ontologies, and knowledge model for the global community.
- To develop classifications of space objects (spacecraft, orbital debris, etc.).
- To provide a dynamic reference model that can be extended by more specific ontologies.
- To be used and further developed with academia, industry, and government agencies such as NASA, the DoD, and international standards organizations

- To contribute to conceptual, terminological and definitional development of space-related of standards, and to contribute to resolving related questions and challenges in space policy and law.

- To explore various applications and innovative ideas of knowledge-based, semantic technology, and artificial intelligence for astronautical disciplines.

The SSAO expands the ideas presented in "An Ontological Architecture for Orbital Debris Data"[10] by Rovetto, published August 2015 in *Earth Science Informatics*. The ontology of SSA, and an early version of the SSAO, is discussed in "Preliminaries of a Space Situational Awareness Ontology,"[8] by Rovetto and T.S. Kelso in *Advances in Astronautical Sciences* (2016). A 2019 paper, accessible via NASA's NTRS server, summarizes the overarching project: "Orbital Debris Ontology, Terminology, and Knowledge Modeling" (R.J. Rovetto, T.S. Kelso, D.A. O'Neil)[13] in the 1st International Orbital Debris Conference, and the Journal of Space Safety Engineering. The overarching Orbital Space Environment Domain Reference Ontology project is currently described at <https://purl.org/space-ontology> and <https://ontospace.wordpress.com/>,[41] with links to GitHub pages to be developed further.

The latest OWL version of the SSAO is tentatively expected to be located at https://purl.org/space-ontology/ssao.[36] Other serialization formats, such as Turtle, with different file extensions would similarly be accessible. Related ontologies, such as the ODO, would have similar permanent links such as <https://purl.org/space-ontology/odo>.[37] The SSAO—and the wider project—is a living, in-progress, ontology effort. It is open to partnerships, contributors, community participation, sponsors, and support for sustainable development.

### 10.4 Scope

The SSAO aims to formally represent data and our dynamic knowledge about the orbital space surrounding a celestial body such as Earth. This includes capturing the meaning of that data, as well as its origins, and how we acquire it. The scope of this ontology is that of the SSA or space domain, aiming to capture common knowledge in this universe of discourse. At the instance level, this encompasses the near-Earth environment and the physical phenomena therein, including artificial satellites (e.g., the International Space Station), their inter-relationships, how they interact; data at a particular time (and over time); and the activities by which we attain information and situational awareness of orbital space. At the generic level, this involves capturing the orbital and near-space environment of any given astronomical body. Combined, this will yield a domain knowledge model, reference model, semantic model, knowledge graph, etc. The scope, more precisely, includes the following:

- The observation, detection, tracking, identification of artificial and natural objects in orbit; as well as the determination and prediction of their past, present, and future position and motion.

- Orbital debris and spacecraft, their parts, properties, behaviors, etc.

- Classes of orbit and the orbits of objects.
- The infrastructure involved in monitoring and modeling the motion of those orbital objects and orbital space.
- The agents (persons, organizations) involved in these activities toward achieving actionable SSA.
- The physical and social infrastructure agents use in SSA-relevant activities or space missions. Examples: sensor networks; data processing systems; ground, launch, and space systems and segments.
- Data: observation, tracking and other sensor data, spacecraft telemetry, etc.
- Data flow processes, e.g., from sensor to spacecraft operator, data analysis activities.
- The observation and predictive activities themselves as well as the resultant data and relevant scientific principles.
- Events, actions, and processes occurring in orbit, e.g., collisions and spacecraft maneuvers.
- Predictions and beliefs about current or future space object positions and trajectories.
- Concepts and scientific principles of astrodynamics, orbital dynamics, etc.

### 10.5 Classes, Object Properties, and Data Properties

This section provides documentation of selected OWL Classes, Object properties, Data properties, and Named Individuals. These are used to express categories, relationships, and actual particular things, respectively. Individuals are specific members or instances of a class. Finally, Annotation properties provide metadata semantic content for the ontology and for particular ontology elements.

RDF(S), OWL, and the following Dublin Core metadata elements are used: rdfs:comment, rdfs:label, dc:title, dc:description, dc:creator, dc:rights. The annotation, rdfs:label, presents the display label (preferred label) for the ontology element. Some author-created annotations are: alternative label, example, definition (of label), logical definition, clarifying note, scope, and development status note. Other vocabulary and metadata resources may also be reused.

Alternative label annotations present synonymous or equivalent labels (terms) as the rdfs:label. The Example annotation provides an actual example or instance of the class in question, e.g., the International Space Station for the class 'Astronautical Habitat' or 'Space Station.' This is helpful where no definition is possible or where a definition by example is desired. Specialized definition and description annotations are also asserted to provide more precise, descriptive metadata for the contents and constructs of the ontology. Formal or logically coherent and consistent definitions are also provided to facility queries and afford machine inference capability (automated reasoning).

Formatting for the ontology constructs is as follows. Labels may be lowercase and written consistently with natural language. The corresponding ontology class, property, or individual will have an IRI short name (a fragment of the full IRI) displayed in natural language with underscores separating composing words. However, as development continues, alpha-numeric globally/universally unique identifiers (GUIDs/UUIDs) for IRIs are anticipated. Class IRI short names are upper camel case (Example_Here), and Object/Data properties are lower camel case (example_Here). When assigning labels, the case of the letters should not be of consequence except for proper nouns, but labels should be comprehensible to human users, and a consistent format is helpful. Named individuals (instances of classes), for example, will be capitalized in accord with the rules of the given natural language. Table 2 presents an example of ontology formatting.

Table 2. Ontology formatting exemplar.

| **Class Example** | |
|---|---|
| Class (IRI short name): | Artificial_Satellite * |
| Example full IRI: | https://purl.org/space-ontology/Artificial_Satellite<br>https://purl.org/space-ontology/5c5a150f_729a_487d_992b_7ce4504cc2b6 |
| Class label (term): | Artificial satellite |
| Super-class: | Satellite |
| Definition/description<br>(of class label or element): | A satellite created by persons.<br>This class is intended to categorize various kinds of satellites made by persons. The (sub)classes, group individuals with properties characterizing the given class. |
| **Example of an Object Property and Data Property** | |
| Object/data property<br>(IRI short name):** | has_COSPAR_Number<br>has_COSPAR_Number_value |
| Label (term): | has COSPAR number |
| Super-property: | has_Identifier |
| Description: | A relationship between an object in orbit (satellite) and the COSPAR identification number assigned to that object. |
| Domain: | Satellite, Orbital_Object, Resident_Space_Object. |
| Range/datatype/value: | COSPAR_number, “1998-067-A” |
| Example formal statement: | International_Space_Station has_COSPAR_Number “1998-067-A” |
| **Example of Named Individual** | |
| Label:*** | International Space Station |
| Description: | The habitable astronautical station orbiting Earth, used for scientific investigation, and constructed in a multi-national effort originating in the 1980s. |
| Is an instance of / is of type: | Space station, Astronautical station/habitat, Artificial satellite |
| Has orbit type/regime: | Low Earth orbit |
| Has international designator: | 1998-067A |
| Has perigee value/distance: | 423.6 |
| Has unit: | kilometers/km |

* Or an opaque identifier such as an GUID, e.g., 5c5a150f_729a_487d_992b_7ce4504cc2b6

** Depending on how the properties or relations are modelled, a data property would have specific value as its range, and the Object property would have a class. Instances of the class, in turn, may be related via another property to a value.

*** Named individuals include labeled orbital debris fragments, which is a critical part of space situational awareness.

Linking annotated ontology constructs in this way creates a knowledge model and a graph structure. Like other OWL ontologies, it expresses knowledge statements in RDF subject-predicate-object expressions that are interconnected. This format, including the explicit formalization of these elements, allows for computation. Classes and their properties are used in queries and rules for automated inference.

For example, the following approximate SWRL rule includes classes Two-line_Element_Set, Satellite, and Sensor_Device; and relations (object or data properties) has_Orbit, described_by, and was_observed_by. The carrot symbol, '^,' signifies the conjunction operator (and). The symbol '->' signifies the conditional operator (if…then):

```
has_Two-line_Element_Set(?s, ?t) ->
     Satellite(?s) ^ has_Orbit(?s, ?o) ^
     described_by(?o, ?t) ^ was_observed_by(?s, ?x)
     ^ Sensor_Device(?x)
```

Read as:
"If *s* is a satellite, *s*, has a two-line element set (TLE), *t*, then *s* has some orbit *o*, and *o* is described by *t*, and s was observed by sensor *x*."

With other rules and axioms, it is intended to express the general knowledge, or domain fact, that satellites with a TLE were observed by some ground or space-based sensor and have an associated orbit that is described by that TLE.

In a development and editor tool, such as Protégé and those listed in section 2.2., computable class expressions, constraints, and axioms that use the vocabulary of the ontology can be asserted. For example, in the Manchester OWL syntax, the SSAO can describe a celestial orbit in the following manner.

A heliocentric orbit class can be qualitatively described as being a type of gravitationally-bound orbit with the sun as the central body:

```
'gravitational orbit' and ('has Central Body' value Sol)
```

In this example, Sol (the sun) is a named individual (a member or instance of the class of Star), and 'has Central Body' is the label for the associated OWL Object Property that relates a class with an individual.

Similarly, the category of Elliptical Orbit can be partially described as being a gravitationally-bound orbit with an eccentricity between 0 and 1:

```
'gravitational orbit'
     and ('has orbital eccentricity value' some
     xsd:decimal[> 0.0 , < 1.0])
```

Here, 'has orbital eccentricity value' is an Data Property displaying a numeric value. That value is of the Decimal datatype and can be within the range of 0 and less than 1.

Finally, the logical properties of relations (object/data properties) can be specified, e.g., symmetric, transitive, reflective, functional (taking one value), etc.

Tables 3 and the two tables that follow list selected Classes and Object/Data Properties with descriptive metadata. Descriptive information is included optionally but may display candidate definitions of the OWL Class or Property label (term), a description of the purpose or intended meaning for the ontology element, (in)direct super-classes separated by commas, and the domain and range of the property (relation). All classes in OWL are subclasses of owl:Thing. Some of this information is presented using specific annotations properties, e.g., rdfs:comment, and user-defined 'definition,' marked in italics. Each row describes a single class or object/data property.

Table 3. Classes.

| Class Label | Description | Super-classes |
|---|---|---|
| Aerospace vehicle | A vehicle designed to operate and travel through both atmospheric and astronomical environments.<br>A vehicle designed for flight in both planetary atmospheres and extraplanetary environments. | Artificial space object vehicle |
| Albedo | A ratio measure of the reflected light, from a celestial body, to the amount incident on that body. | Physical property |
| Apoapsis | The point, in an orbit, most distant from the central body. More precisely, the point in orbit most distant from the apocenter of the orbital system. | Orbital property, spatial location, spatial point, geometric point |
| Apoapsis distance | The distance from an orbiting object at apoapsis to the orbital system apocenter. | Orbital property, spatial extent |
| Apogee | The point, in an orbit, farthest from the Earth. | Orbital property, orbit part, spatial location, spatial point, spatial position |
| Argument of periapsis | The angle, measured in degrees and represented by Greek lowercase omega ($\omega$), from the ascending node of an orbit to periapsis. | Orbital property, spatial extent, spatial area, geometric distance |
| Argument of perigee | The argument of perigee, represented by Greek lowercase omega ($\omega$), describes the orientation of an orbit in its orbital plane relative to Earth. It defines where the low point, perigee, of the orbit is with respect to the Earth's surface.[44] It is the angle, measured in the direction of satellite motion, from the ascending node to perigee. It has values greater than 0 but less than 360.† The angle from line of nodes to satellite position vector. Its value is undefined when orbital inclination is 0° or 180°, or when orbital eccentricity is 0. | Classical Keplerian orbital element,<br><br>Direct super-class: argument of periapsis |
| Artificial satellite | A satellite created by persons. An artifact intentionally created by persons to operate in orbit and perform one or more functions.<br>Alternative labels: satellite device, artifactual satellite | Satellite |
| Artificial sensor | A technological (artificial/artifactual) system that transmits and/or receives electromagnetic energy. | Sensor |
| Artificial space object | A space object that was created by persons or agents, i.e., not created through natural processes. | Space object |

† These quantitative details can be formalized, in the ontology, as constraints on the ontology constructs.

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Asteroid | Development Note: may be imported from an astronomy source or ontology/vocabulary. | Astronomical object |
| Astrodynamic process | A class categorizing spaceflight activities/processes. | Owl:Thing |
| Astronautical craft | A craft (artifact or artificial system) designed for spaceflight.<br>An artificial system designed to operate in the microgravity environment of astronomical/outer space.<br>Development note: disambiguate from astronautical system, vehicle, and vessel. May be equivalent to astronautical system.<br>Description of ontology construct: This class is intended to categorize various kinds of technological or engineered systems specifically or exclusively designed to operate in outer space. It encompasses vehicles for transporting persons and cargo; unmanned spacecraft, extra-planetary habitats such as space stations, etc. | |
| Astronautical system | A technological system (artificial system, or artifact) designed to operate in the microgravity environment of astronomical space.<br>Alternative label: spaceflight system | Artificial system |
| Astronautical vehicle | A vehicle designed for spaceflight transportation, i.e., for transportation (and other operations) in astronomical space.<br>Development note: Disambiguate from 'astronautical vessel'.<br>Alternative label: spaceflight vehicle | Spacecraft, astronautical craft, astronautical system, astronautical vessel |
| astronautical vessel | An artificial system/craft or vessel designed to carry persons or payloads in outer space. Not necessarily for transportation, i.e., not necessarily with a propulsion system. May include astronautical escape pods, space shuttles … | Spacecraft, astronautical craft |
| habitable astronautical vehicle/vessel/system | An astronautical vehicle capable of supporting life, i.e., a habitable spacecraft.<br>Alternative label: habitable space system, astronautical habitat<br>Example: The International Space Station | Astronautical system |
| Astronomical event | An astronomical event is a naturally-occurring event in astronomical space involving astronomical bodies. | Event |
| Azimuth | – | – |
| Break-up event | A destructive event in which an orbiting object-fragments. | Orbital-event |
| COSPAR number | An international designation, assigned by the Committee on Space Research (COSPAR), to objects launched into outer space.<br>Alternative labels: International Designator, NSSDCA ID | Identification number, identifier, space object identifier |
| Cartesian ephemeris | – | Ephemeris |
| Catalog updating process | The process of updating a catalog or data-base (e.g., space satellite catalogs) with new data. | Cataloging activity |
| Celestial body | A physical body or object in the astronomical space environment.<br>Examples: Earth, stars, asteroids.<br>Alternative labels: celestial object, astronomical body | Physical object |
| Central body | The most massive of the bodies in an orbital system, exerting the dominant gravitationally-binding force for its satellites.<br>A central body is considered relative to a given orbital system.<br>Example: the sun is the central body of our solar system.<br>Development note: can be ontologically modeled as a role played by a celestial body, e.g., as a body that is assigned the role of central body by an observer. | |
| Circular orbit | An orbit with an eccentricity of zero.<br>It is a special case of elliptical orbits. | Gravitational orbit |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Classical Keplerian orbital element | The orbital elements are the properties, features, measurable quantities, or parameters required to identify an orbit, describe the motion of a satellite in orbit. Six parameters are needed to uniquely define an orbit and a satellites position in orbit. They are parameters for describing an orbit, for describing non-inertial orbital trajectories of the orbiting bodies.<br><br>The Keplerian (or classical) elements describe the size, shape and orientation of an orbit, as well as the location of satellite in the given orbit.<br><br>An orbital element set is a set of quantities specified relative to a coordinate system and reference frame to describe the orbit of a satellite. Position and velocity (two elements) are needed to define an orbit. With these elements, a satellites past and future position and velocity may be predicted. Three parameters for each are needed to describe position and velocity in three-dimensional space. Thus, six parameters are needed for an element set. | Orbital property |
| Collision avoidance maneuver | An orbital maneuver performed by a spacecraft in order to avoid colliding with another object.<br>An orbital maneuver to alter the orbit of a spacecraft in order to maintain close approach and conjunction assessment criteria. | Spacecraft orbital maneuver |
| Collision plane | – | – |
| Communications satellite | An artificial satellite designed to provide communications between agents, i.e., that is given a communications purpose/function. | Artificial satellite, spacecraft |
| Search and rescue satellite | An artificial satellite designed to support search and rescue (SAR) operations, i.e., whose purpose/function is to support SAR operations. | Artificial satellite |
| Conjunction assessment process | The process of assessing or determining the closest distance between two orbiting objects (or a launch vehicle), i.e., determining the point of closest approach to assess likelihood of collision.<br>The computational process of estimating the probability that two or more objects will come within a specified distance from one another, and thereby present a risk of collision. | SSA process,<br><br>Computational process |
| Coordinate system | A system of coordinates to identify the position of a point, object, or direction.<br>Clarifying note: they are specified on a given reference frame in order to determine position and motion of an object. | Reference entity, mathematical / geometric construct |
| Covariance matrix | – | Data object, information object |
| Data format | A format for presenting data in a structured manner. | Owl:Thing, information object |
| Data source | The origins of a dataset, data element, or database.<br>A type of provenance data. In SSA, it's important to know the source of data, in part, to understand its quality, and to compare the data with multiple data sources toward more complete, accurate and actionable SSA. | Owl:Thing |
| Defunct spacecraft | Spacecraft that are no longer functional, or incapable of being reactivated. | Orbital debris |
| Fragmentation debris | Debris objects that were formed by processes such as collisions, shedding, surface deterioration, and explosions. They are portions of a larger object and were unintentionally separated from the larger whole. | Orbital debris |
| Mission-related debris | Debris objects that have been discarded or lost during normal mission operators, rather than due to anomalous events such as collisions. | Orbital debris |
| Docking maneuver | A spacecraft maneuver in which a spacecraft approaches and connects to another, i.e., in which a spacecraft engages in a docking activity or process. | Proximity operations maneuver |
| Earth-centered inertial frame (J2000) | A reference frame with Earth's center as the origin.<br>The x-axis points toward the vernal equinox at Epoch J2000. The y-axis is in the Earth-Sun equatorial plane. The z-axis points to the celestial north pole at Epoch J2000. | Coordinate system, reference system |
| In-track direction/axis | – | – |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| International celestial reference system | The origin of the ICRS is the barycenter of Earths solar system.<br>The x-axis is points to the vernal equinox at epoch.<br>The y-axis is in the Earth-Sun equatorial plane.<br>The z-axis points to the Earth's celestial North Pole at epoch. | Coordinate system,<br><br>Reference system |
| Ephemeris | An time-ordered list or table of the predicted position and velocity (data) about an orbital object. In SSA it is generated with space object tracking data and computational models. | Information object, data object |
| Epoch | A specified point in time. It is a specific point in time used as a reference. It is a time when a dataset is valid. The temporal reference frame. The time at which the orbital elements were taken/ acquired. | Orbital property |
| Exclusion ellipsoid | A specified ellipsoid volume around an orbital object within which is considered a close approach or potential conjunction. Violation of this 3D distance around the object indicates a risk of collision, resulting in further data collection and conjunction assessment analyses.<br>Alternative labels: exclusion volume, keep-out volume, screening volume | |
| Orbit | The path described by one body in its revolution around another (Merriam Webster's Dictionary).<br>Development note: ongoing conceptual development. May be characterized in various ways, e.g., as an extent, path, motion, process, relational object, etc. | Owl:Thing |
| Gravitational orbit | The path described or traced by a physical body periodically revolving around another body due to gravitation. A curved path described by an object under the influence of gravitation.<br>Gravitational orbits are orbits bound by gravitation and determined by the combination of gravity and the motion of the orbiting object.<br>They are gravitationally-bound orbits.<br>The path of an object orbiting a gravitationally-binding mass. An orbit within a sphere of influence.<br>The gravitationally-induced path or motion of an object revolving around a gravitationally-binding entity. The path traveled by an object bound by gravitation around another object. The periodic path or motion of an entity (the orbiter) around another entity (the orbited / central entity / central body) due to gravitation. | Orbit |
| Geosynchronous orbit | GEO is an Earth Orbit of an approximate altitude of 35856 km. | Gravitational orbit |
| Equatorial orbit | A Low Earth Orbit that has an orbital inclination between approximately 0° and 20°. | Gravitational orbit |
| Elliptical orbit | An orbit with eccentricity greater than 0 but less than 1. Alternative label: elliptic orbit, Kepler orbit | Gravitational orbit |
| Heliocentric orbit | An orbit in which the Sun is the central body. | Gravitational orbit |
| High Earth orbit | An orbit of approximately 35,780 km altitude. | Gravitational orbit |
| Hohmann transfer orbit | An elliptical transfer orbit used by a spacecraft to move from one circular orbit to another, within the same orbital plane, and conducting two orbital maneuvers (or burns). | Transfer orbit, Elliptical orbit |
| Inclined orbit | An orbit that is has an inclination other than 0°. | Gravitational orbit |
| Low Earth orbit | Note: There are multiple definitions of 'LEO' , often with differing altitudes, but in general a low Earth orbit is an Earth orbit below approximately 2,000—5,000 km altitude. | Gravitational orbit |
| Human spaceflight | Spaceflight in which human beings participate.<br>Alternative labels: manned spaceflight | Spaceflight |
| Launch date | The date a spacecraft, satellite, payload or launch vehicle was launched. | |
| Measurement value | A specific numeric quantity resulting from a measurement activity and having a unit of measure.<br>Alternative label: measurement | |
| Mean anomaly | The average of the true anomaly. | Orbital property |
| Metric data | Data about the position of an object. It includes azimuth, elevation, range, range rate, and time.<br>Alternative label: observation | SSA data |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Miss distance | The distance from which one orbital object will pass another. A prescribed distance within which an orbital object would be considered a potential collision risk. It is used in conjunction assessment processes.<br>A prescribed distance, relative to a target resident space object, from which other orbital objects should not violate. | Orbit determination concept |
| Micrometeoroid | A naturally-formed astronomical object | Astronomical object |
| Near-Earth object | Alternative label: NEO | Astronomical object, near-space object |
| NORAD catalog number | A five-digit number assigned by the North American Aerospace Defense Command (NORAD) to uniquely designate each satellite in their catalog.<br>Alterative label: NORAD catalog ID, NORAD ID, NORAD catalog identifier | Identifier |
| Natural celestial body | An naturally-formed astronomical body. An astronomical body created through natural processes.<br>Alternative label: astronomical body, natural celestial object | Celestial body, Space object |
| Observation | The detection of an object in orbit or through the spatial region being observed.<br>Development note: Disambiguate between the activity of observing, the observer, the observed target, and the result or product of observing, the latter of which is typically a dataset in SSA. | owl:Thing |
| Observation data | In SSA, may be equivalent to metric data. | SSA data |
| Observation dataset | Data that sensor devices record after detecting (or receiving) electromagnetic radiation from their field of view. The typical meaning of an observation. | SSA data |
| Observation process | A process in which a sensor device observes some portion of space by active or passive means. It results in an observation dataset, which is used for orbit determination, propagation, and conjunction assessment activities. | SSA process |
| Operational satellite | An artificial satellite that is operational. | Artificial satellite |
| Orbit regime | A broad category of potential orbits.<br>Example: LEO<br>Alternative label: orbital regime | |
| Orbital conjunction | A conjunction in SSA is a close approach of two tracked objects, which represents a potential collision, according to specified distance criteria.<br>It is when the miss distance between two objects is less than specified criteria.[45] A conjunction event is often described as an event in which the relative separation between objects reaches a specified point of closest approach. | |
| Orbital collision | A collision between two or more objects in a gravitational orbit. | Orbital event |
| Orbital debris | Orbital debris is any man-made object in orbit about Earth which no longer serves a useful purpose (NASA).<br>Debris located in orbit about a celestial body.<br>Description of the class: The Orbital_Debris class is a category of orbital space debris objects. It is subdivided into sub-classes for types of orbital debris, such as fragmentation debris.<br>Orbital space debris can be natural or artificial.<br>Alternative label: orbital debris object | Space debris object |
| Orbital debris field | A collection of orbital debris objects, which may have a common formation event (cause), and may share similar motion, dynamic and physical properties. | Space debris object |
| Orbital debris mitigation | Techniques, activities, methods, or structures designed to limit the creation of orbital debris. | owl:Thing |
| Orbital debris removal process | An activity designed and aiming to remove orbital debris from orbit. | owl:Thing |
| Orbital eccentricity | The eccentricity of an orbit describes the shape of an orbit, i.e., how much the orbit deviates from a circle. It is represented with the symbol “e”. Closed orbits have an eccentricity ≥ 0 but less than 1. Open orbits have eccentricity ≥ to 1. Eccentricity is the ratio of half the foci separation to the semi-major axis. For elliptical orbits, it can be calculated by the apoapsis minus periapsis divided by twice the semi-major axis (or the apoapsis plus the periapsis). | Classical Keplerian orbital element |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Orbital event | Events occurring in an orbit about a celestial body, and in which a natural or artificial object participates. | Owl:Thing |
| Orbital inclination | Orbital inclination describes the tilt of the orbit or orbital plane. It defines the orientation of the orbit with respect to the Earth's equator. It is represented with the symbol 'i'. Equatorial inclinations are 0° or 180°. Prograde are greater or equal to 0° but less than 90°. Polar orbits are 90°. Retrograde orbits are greater than 90° but less than 180°.<br>The class intended to represent inclination of gravitational orbits, orbital trajectories, or objects in orbit. | Classical Keplerian orbital element, orbital property |
| Orbital space object | A physical object located in the orbital space environment of a celestial body, and in orbit about that body. It is an object gravitationally-bound to orbital motion about a celestial body or center of gravity.<br>Development note: This, and related classes, are in the ODO.[10] | Space object |
| Orbital path | A path through space or space-time that an object traverses and describes, over time, in its motion around some astronomical object or center of gravity. The spatial or spatio-temporal path an object traces/follows in orbit about a celestial body or center of gravity. | Owl:Thing, Path |
| Orbital period | The time required for a satellite to complete an orbit (or revolution) about its central body. The unit of measure is typically minutes. | Orbital property |
| Orbital plane | The two-dimensional plane upon which the path of an orbiting object is projected in order to help describe the objects orbit and motion. | Two-dimensional plane, spatial plane |
| Orbital property | A property, feature, parameter, quality, or characteristic of an orbit. This is a class intended to categorize various types of properties of orbits: the classical orbital elements, apoapsis, perigee, etc.<br>Alternative Label: orbit property | Physical property |
| Orbital space environment | The space environment surrounding a central body or entity and in/through which objects orbit that central entity.<br>The space environment, relative to a central entity, in which orbital motion occurs | Space environment, spatial volume |
| Orbital state vector | – | Data format |
| Orbital system | An orbital system is a dynamic physical system in which one or more masses periodically revolve about the systems barycenter. | Owl:Thing |
| Orbital trajectory | A trajectory is, or represents, the spatial and/or spatio-temporal location of an object over time.<br>An orbital trajectory is, or represents, the location of an object in its orbit over time. | Orbital path |
| Orbital velocity | The velocity required to sustain a satellites orbital motion about its central body/point. | Physical property |
| Orbiting process | A process in which an object orbits a center of gravity. | Physical process |
| Owner | An actor or agent who legally owns the device or technological system in question.<br>Development note: specialize further | SSA actor |
| Passivation | The removal of stored energy from a spacecraft. | Owl:Thing |
| Payload | The cargo or contents, to be transported, of a craft or vehicle and which are not required for the operation of that vehicle. | Owl:Thing |
| Perigee | The closest point, in an orbit about Earth, from the surface of the Earth. | Orbital property, periapsis, spatial point, spatial location |
| Radius of periapsis | The smallest distance, from the surface of a central body (e.g. Earth) to the orbiting body, in the orbit of the orbiting body. | Orbital property |
| Physical process | A process that has spatial, temporal, and/or spatio-temporal properties.<br>This is a generic high-level category for various kinds of process. It, together with other abstract concepts, will be grouped in a separate ontology module to import. | Owl:Thing |
| Physical property | A generic property that has spatial, temporal, and/or spatio-temporal properties. | Owl:Thing |
| Planet | – | Astronomical object |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Planetary orbit | A planet's orbit about some other astronomical body (e.g., a star) or center of gravity.<br>Contrast with 'planet-centric' or 'planetocentric' orbit, defined as the orbit of a body about a planet. | Gravitational orbit |
| Primary satellite | Signifies the object of interest (or focus) in conjunction assessment. It is the object relative to which another orbital object is evaluated for risk of collision.<br>Development note: The concept of a primary (and secondary) satellite in conjunction assessment can be modeled as ontological roles played by a physical object in orbit.<br><br>An object is primary relative to the observer and their activity of assessing its likelihood of colliding with other objects.<br><br>Clarifying note: Similarly, 'primary body' in celestial mechanics typically refers to the celestial body, in an orbital system, with the largest mass.<br>Alternative labels: primary object | Orbit determination concept |
| Probability of collision | The probability that two objects, or their centers of mass, will come within a specified distance from one another at time of closest approach. | Orbit determination concept |
| Propagation process | The propagation process in an SSA context involves the prediction of spacecraft future positions by considering past and current observations (observational data). | SSA process |
| Propulsion subsystem | A subsystem designed to propel a vehicle forward.<br>Alternative label: propulsion system | Spacecraft subsystem |
| Protected orbit | – | Gravitational orbit |
| Proximity operations maneuver | A spacecraft maneuver in which one spacecraft navigates in relative close quarters to another. | Spacecraft maneuver |
| Reference body | A physical object (e.g., planet, spacecraft, etc.) used to fix and orient a coordinate system. | Reference entity |
| Reference direction | A part of a reference frame beginning from the reference origin. | Reference entity |
| Reference entity | An entity—singular or collective—in relation to which measurements, calculations, observations, or other activities are conducted. | Owl:Thing |
| Reference frame | – | Reference entity |
| Reference origin | A part of a reference frame, through or on which a coordinate system or set of orthogonal axes is projected. It is the point of intersection of those axes. A point (either mathematical or physical) used as a reference in defining or fixing a coordinate system. | Geometric reference entity, spatial point, spatial location |
| Reference plane | A 2D plane associated with some body, and part of a reference frame used in defining a coordinate system | Geometric reference entity |
| Rendezvous maneuver | A proximity operations maneuver in which one spacecraft navigates to physically connect with another | Spacecraft orbital maneuver |
| Resident space object | An object residing in orbit about Earth or another celestial body. A space object in orbit or orbital motion about Earth or another celestial body. A space object orbiting another object.<br>Clarifying note: This term is sometimes used to signify artificial Earth-orbiting objects. However, the term itself does not indicate the artificial or Earth qualification. A more precise term for that sense is recommended. Further conceptual and semantic analysis needed. | Space object |
| Right ascension of the ascending node | Describes the rotation of the orbital plane about a central astronomical body such as Earth. It is the angle, measured eastward, from the vernal equinox to the ascending node. It defines the location of the ascending and descending orbit locations (nodes) with respect to the Earth's equatorial plane.[44]<br>It has values greater or equal to 0° but less than 360°, and undefined for equatorial orbits. It is a measure of the angle from the x-axis of the reference coordinate system to the line of nodes.<br>Example: the angle from vernal equinox direction to the line of nodes. Alternative label: RAAN | Classical Keplerian orbital element |
| Rocket body debris | A category of orbital debris that includes rocket body parts, such as rocket body stages. | Orbital debris |
| Orbit semi-major axis | The distance from the center of an orbit to the apoapsis or periapsis. | Orbital property |
| SSA actor/agent | An actor (person, group, organization, agency, government, nation-state, etc.) engaging in SSA activities or using SSA services or data. | Owl:Thing |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| SSA data | Data used to attain space situational awareness, or data about objects in orbit. Clarifying note: Ground- and space-based sensors receive signals from objects in orbit. Those signals are recorded and collected as data, which is processed in order to track, identify and predict the future position of the object. | Owl:Thing |
| SSA event | An event unique to SSA or involve in SSA activities. | Owl:Thing, Event |
| SSA facility | A building or structure that supports SSA. | Owl:Thing, Facility |
| SSA message | A message sent via computer or signal by an SSA facility, network, sensor device, computer system, or agent. | Owl:Thing, Notification message |
| SSA network | SSA Networks are composed of ground and space-based sensors, control stations, and other aspects.<br>An SSA Network is an artificial network with a function to achieve or maintain space situational awareness, and whose network parts include sensors, facilities, communications channels, etc. | Owl:Thing |
| SSA process | An activity performed by a space situational awareness (SSA) network or SSA actor.<br>SSA includes data collection, data storage (for cataloging objects), and data processing (for predicting motion) from observations of object in outer space (orbital space and deep space environments). | Owl:Thing |
| SSA stakeholder | An agent that has an interest in, or gains value in, SSA. | SSA actor |
| Satellite name | The name of a satellite expressed as an string of alphanumeric characters. The proper name for a satellite. | Identifier |
| Spacecraft operator | – | SSA stakeholder |
| Screening process | The computational process determining which objects will come within a specified distance of a primary satellite. The screening process is composed of multiple activities, such as filtering objects. It may result in the release of a conjunction data message. | Orbit determination concept |
| Search and rescue satellite | An artificial satellite whose function or mission is to support search and rescue operations. | Artificial satellite |
| Sensor network | A network of distributed but interconnected sensors. | Owl:Thing |
| Sensor observation | A sensor observation (dataset) is the position (possibly also velocity) of an object in outer space taken at a particular time and relative to the observing sensor.<br>Modeled as a set of recorded quantities, a particular observation is an instance. | Observation |
| Sensor tasking process | The process of commanding a sensor device to adjust its position or take specific observations. This includes the activities of sending messages to the sensor device. | SSA process |
| Space-based sensor | A sensor that resides and is designed to operate in the microgravity environment of outer space. | Artificial sensor |
| Space agency | A national organization conducting, managing or coordinating spaceflight or astronautical activities and missions. | SSA actor |
| Space body frame | – | Coordinate system |
| Space debris object | Natural or artificial objects located in outer space, and which are not usable or do not have a function.<br>A natural or artificial physical object in astronomical space that, by virtue of some event or process, has lost its function, or has ceased to be a functional or structural part of its whole. | Space object |
| Space domain | If broadly construed, this is equivalent to the astronomical domain, i.e., all objects and phenomena in outer space. Some contexts use the term to signify the space objects, environment and actors of interest relative to Earth, to a nation-state, and/or gov't agency. Some have used it to signify the SSA domain. A more precise term would be 'orbital space domain' if intending to mean the orbital environment. | Owl:Thing |
| Space environment | Alternative label: astronomical environment, microgravity environment.<br>The outer space environment. | Owl:Thing |
| Space mission | Alternative term: spaceflight mission, astronautical mission. | Owl:Thing |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Space mission plan | Disambiguate from space mission definition, space mission operations concept, etc. | Specification |
| Space mission phase | Part of the entire activity/process of a space mission and consisting of sub-activities unique to or characteristic of the given phase.<br>Part of a space mission, where space missions are described as a temporally occurring entity. | Owl:Thing |
| Space mission segment | An astronautical mission part that includes persons, artificial systems, and facilities supporting the execution of the mission. A partition or grouping of the systems, personnel and processes necessary to realize a space mission, e.g., ground segment, space segment, launch segment, user segment. | Owl:Thing |
| Space object | An object located in outer space.<br>Development note: too vague a term. use more specific categories instead. | Owl:Thing |
| Space object catalog | A data repository (e.g., a database) housing data about objects in the orbital space environment, in astronomical space, or outer space.<br>Alternative label: space object database | Owl:Thing |
| Space object contact | A category intended to signify a detected object in the orbital space environment. | Owl:Thing |
| Space object identification process | The process of identifying a space object, i.e., an object detected in astronomical space. | SSA process |
| Space object identifier | An alphanumeric identifier or designation assigned to a particular detected space object. | Identifier, Identification number |
| Space object role | A role played by an object located in outer space. Note: needs further conceptual development.. | Owl:Thing |
| Space object tracking process | The process or activity of tracking (or following the position) of a space object over time | SSA process |
| Space segment | A space mission, and the SSA domain, is often described as having segments: the ground segment, control segment, launch segment, space segment, user segment, etc. Each represents a grouping of systems, activities, and personnel that form part of and realize the overall mission, or SSA system. | Space mission segment |
| Space shuttle | A reusable spacecraft designed to carry persons and cargo. Development note: Disambiguate from STS. | Spacecraft |
| Space situational awareness | SSA (relative to Earth) is awareness, or knowledge, of the orbital and near-Earth space environments. This involves observing, understanding and predicting the motion of objects in that environment, how they are related to the environment and each other, how they interact and behave, and what their properties are. It is essential to understand the causal relationships among near-Earth objects, space weather, spacecraft and orbital debris.<br>SSA can be conceived as the understanding derived from studying the near-Earth space environment (T.S. Kelso).<br>The SSA domain includes natural celestial bodies (e.g., near-Earth objects), space weather phenomena, spacecraft, orbital debris, and the processed by which we observe and gain knowledge of those entities.<br>An essential task of SSA is observing, detecting, tracking identifying and predicting the motion of objects in orbital space, i.e., space surveillance.<br>Achieving and maintaining SSA requires orbit determination (identifying where an object is now), orbit propagation/prediction (identifying where an object may be in the future), and assessing risk of collisions. | Owl:Thing |
| Civil space situational awareness | – | Space situational awareness |
| Space station | A spacecraft designed to be habitable for extended periods of time. It typically does not have a means of propulsion, aside perhaps from attitude or maneuvering thrusters for station-keeping. In other words, it is not typically designed for self-propelled travel.<br>Alternative label: astronautical station | spacecraft, astronautical habitat |
| Space surveillance | The activity of observing, detecting, cataloging, tracking, and identifying objects in outer space. | SSA process |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Spacecraft | A technological craft or system designed to operate in the microgravity environment of astronomical space. | Artificial space object |
| Unmanned spacecraft | A spacecraft that is not designed to carry persons. | Spacecraft |
| Space vehicle | A spacecraft designed for self-propelled flight in the outer space environment / in the microgravity environment / in astronomical space.<br>Alternative label: spaceflight vehicle, space transportation vehicle, astronautical vehicle | Spacecraft |
| Spacecraft part | A spacecraft part is a part of a spacecraft and may include spacecraft subsystems or components thereof. | Owl:Thing |
| Spacecraft bus | The physical framework or platform for the structure of spacecraft, and upon which a devices can be affixed and secured. | Spacecraft part |
| Spacecraft component | A component of a spacecraft.<br>Clarifying note: components are understood as a subclass of 'part', i.e., an engineered part necessary to achieve an objective, or contribute to the functioning of the system it is part of. | Spacecraft part |
| Spacecraft constellation | This class is intended to represent kinds of organized collections of spacecraft that may have a common mission, function, design, or that may achieve a mission only if working together. | Owl:Thing |
| Spacecraft operator | An actor (or agent) having the authority to operate, and responsible for the safe operations of, a spacecraft. | Operator |
| Spacecraft owner | An actor or agent who legally owns the spacecraft in question. | Owner |
| Spacecraft orbital maneuver | An orbital maneuver performed by a spacecraft. A maneuver, in orbit, performed by a spacecraft and directed either autonomously or via command by a remote controlling station or operator. | Orbital event |
| Spacecraft payload | The payload of a spacecraft. | Spacecraft part |
| Spacecraft position | The position of a spacecraft at a specific time. | Owl:Thing |
| Spacecraft state | The aggregate of the position and velocity of a spacecraft at a given time.<br><br>Spacecraft position and velocity at a specific time are collectively referred to as the state of a spacecraft. | Orbit determination concept |
| Spacecraft subsystem | A subsystem of a spacecraft. | Spacecraft part, component |
| Spaceflight | The activity or process of flying in the microgravity / outer space / astronomical environment.<br>The controlled motion of a spacecraft in outer space. | Owl:Thing |
| Spaceflight operation | An operation (a deliberate and coordinated activity) performed by agents and/or spacecraft during a spaceflight mission (astronautical mission) to accomplish some objective or achieve some function.<br>Example: a spacecraft maneuver | Owl:Thing |
| Special perturbations model | A mathematical and computational model used in orbit propagation computations. Alternative label: 'Special perturbations orbit propagation theory' | Orbit determination concept |
| TLE set | Alternative label: TLE, ELSET, elset<br>A data format encoding a set of orbital parameters to describe an orbit at a particular time. A TLE set typically consists of 2–3 lines (line 0, 1, 2) of symbols, groups of which have particular meaning. Line 0 may or may not be present but can include spacecraft name and dimensions. A TLE is generated using the Simplified General Perturbations Theory (SGPT).<br>Clarifying note: The TLE was developed by NORAD (North American Aerospace Defense Command). Development: Disambiguate from the format and the TLE set as a dataset. | Data format |
| Time of closest approach | The time at which two objects orbiting a common central body, and which are believed to be at risk of collision, reach the smallest distance from one another in their orbital trajectories. | Orbit determination concept |
| Timestamp | The time at which a particular event occurs, a particular sensor observation is made, a particular dataset is acquired/recorded, etc. | Physical property |

Table 3. Classes (continued).

| Class Label | Description | Super-classes |
|---|---|---|
| Ttracking station | A station (a type of facility) that tracks the motion of a given object. Alternative label: tracking facility | SSA Facility |
| Unmanned spacecraft | A spacecraft not designed to carry persons. | Spacecraft |
| X-axis | The x-axis (part) of a coordinate system. | – |
| Zenith | – | – |

Figures 25–27 display graph visualizations of portions of the class hierarchy. The OntoGraf plugin in Protégé, as well as GraphViz 2.38, are used to render the diagrams. Rectangular shapes represent classes. Arrows are labeled with the respective object/data property that represent relationships.

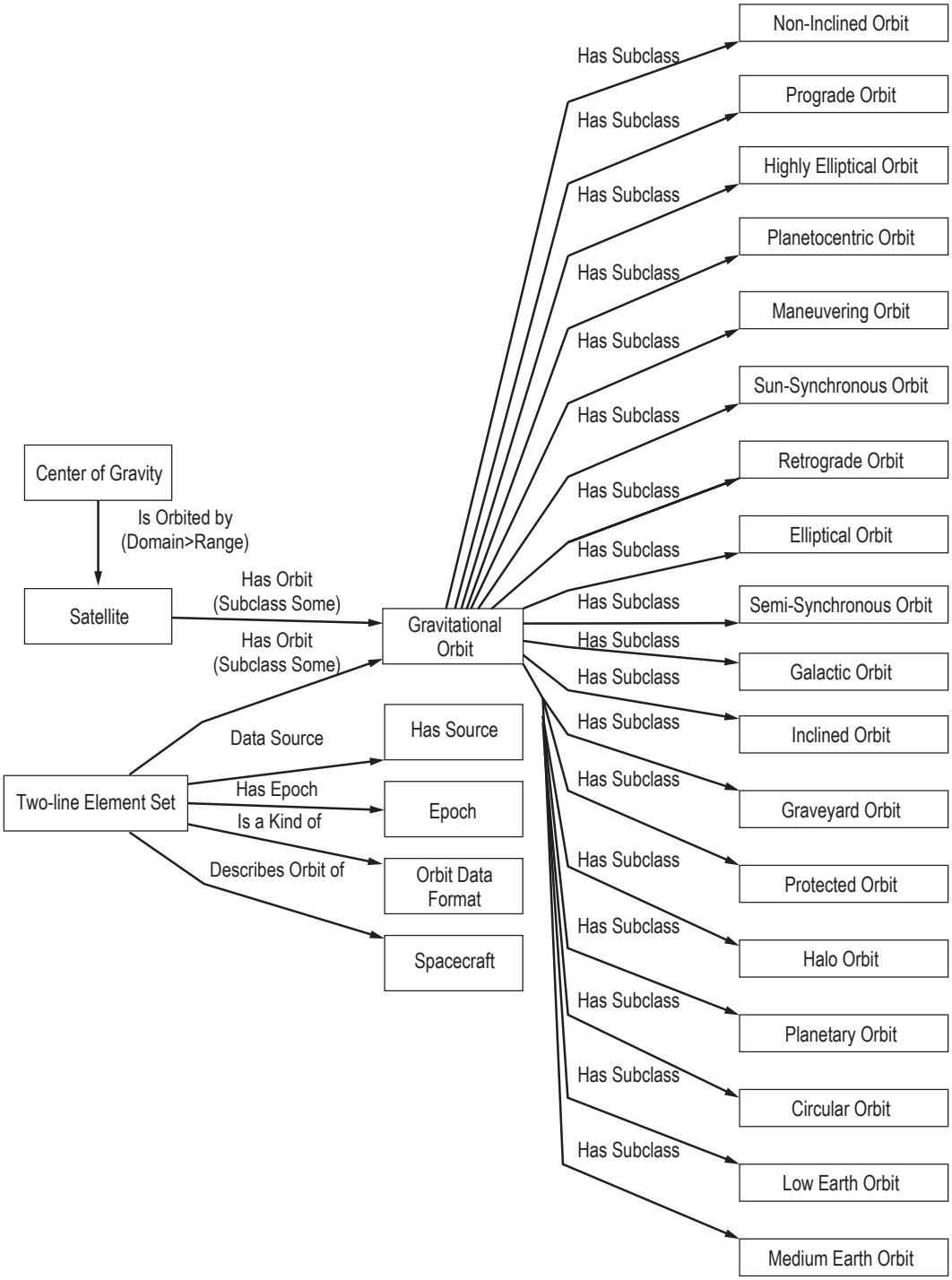



Figure 25. Relationships among satellite, orbit, and TLE classes.

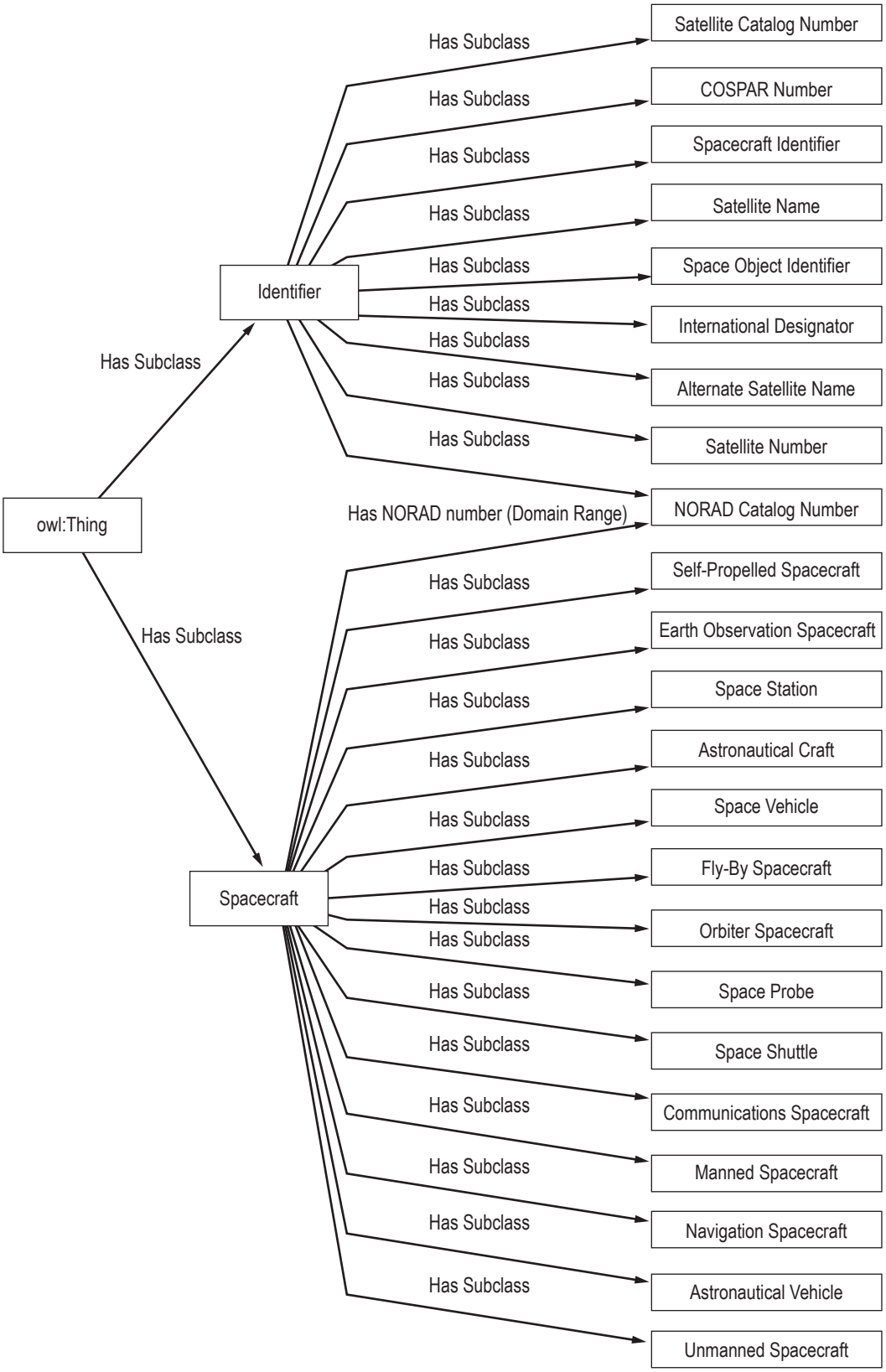



Figure 26. Subclasses of identifiers and spacecraft and a relationship between them.

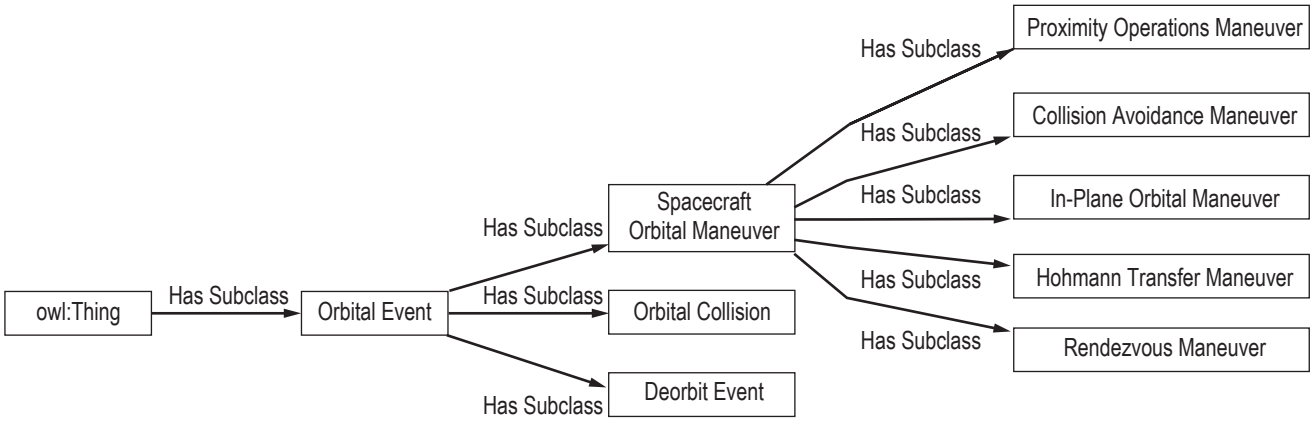



Figure 27. A portion of the Orbital Event class hierarchy.

### 10.5.1 Data Properties

Table 4 lists information about some data properties in the ontology. The domain and range/datatype column includes classes, the instances (individuals) of which are related—via the given data property—to a literal value (e.g., a string of the Datatype xsd:string) as the range. Examples may be provided.

Table 4. Data properties, domain and range, and data types.

| Data Property Label | Description | Domain; Range/Datatype |
|---|---|---|
| Has value | A generic relation between an entity (broadly construed) that can have a range of specific quantitative values, and a particular value of any datatype (integer, float, double, string, numeric, etc.). It groups specialized relations. | |
| Has international designator value | Indicates the international identification number assigned to an orbital object by COSPAR.<br>Alternate Label: COSPAR ID, COSPAR number, COSPAR identifier, NSSDC ID, NSSDC number | Datatype: String (xsd:string)<br>Example: 1997-051J |
| Has measurement value | A generic relation indicating a specific numeric measurement of some physical property or phenomenon.<br>Alternative label: 'has measurement' | |
| Has NORAD number value | This functional data property relates (or links) an object and its assigned NORAD identification number.<br>A specific NORAD #, e.g., 33771, will be displayed. | Orbital object, Space object, Spacecraft, Artificial satellite, …;<br>Datatype: Integer |
| Has spacecraft name | Indicates the proper name of a given spacecraft. | Spacecraft<br>Datatype: String |
| Has operational status value | Indicates whether a spacecraft is active, inactive, non-functional, etc., with a string value. Development Note: may be modeled as an enumeration. | Datatype: String<br>Range of values: "active", "inactive", "defunct", "standby", … |
| Has orbital property value | This data property serves as a generic relation grouping other data properties, such as has_Orbital_Eccentricity_value. It links an orbit, orbital trajectory, or orbiting object, and specific values of orbital properties such as eccentricity, inclination, etc.<br>Alternative label: 'has orbital property measurement' | |

Table 4. Data properties, domain and range, and data types (continued).

| Data Property Label | Description | Domain; Range/Datatype |
|---|---|---|
| Has orbital eccentricity value | Relates an orbit, or orbital object, and a particular quantitative value of the orbital eccentricity at a particular time. Indicates the value of the eccentricity. | Gravitational orbit;<br>Data type: Double.<br>Range of values: between 0 and 1. |
| Has orbital inclination value | Relates an orbit, orbital trajectory, or orbital object, and a particular value of the orbits inclination at a given time. Indicates a numeric value, in degrees, of an orbit's inclination. | Gravitational orbit;<br>Datatype: Double<br>Unit: Degrees |
| Has first time derivative of mean motion value | Indicates a specific value of mean motion for a orbit, or as indicated in an orbit dataset, such as the TLE. The first time derivative of mean motion is one of the parameters included in the TLE set. | Gravitational orbit;<br>Datatype: Double |
| Has radius value | Indicates the numeric value of the magnitude of the position vector of a satellite. | Datatype: Double |
| Has right-ascension of the ascending node value | Alternative label: has RAAN value | Datatype: Double |
| Has miss-distance threshold value | Indicates the specific numeric value for a minimum distance between two orbiting objects, within which would be considered a hazard, i.e., a potential collision risk.<br>Alternative Label: 'has separation distance threshold value'<br>Datatype: Double | |
| Has probability of collision value | Displays the numerical value of probability that two observed objects are going to collide. | Datatype: Double |
| Has unit of measure | Links a quantity, measurement, or physical property quantitative value, with its unit type. It indicates the unit of some quantity or measurement.<br>Example: Meter | |
| Has size | Relates a physical object (e.g., orbital debris fragment) to a value indicating the approximate size of the object. | Datatype: String<br>Example: "small" |
| Is active | Indicates whether the spacecraft is active or not at a given time. | Datatype: Boolean (true or false) |
| Is debris | A relation indicating whether an observed object is a debris object. | Datatype: Boolean |
| Is defunct | Indicates whether an observed orbital object is defunct, i.e., dead or incapable of being reactivated. | Datatype: Boolean |
| Is controlled | Indicates whether the motion of an observed orbiting object is controlled, i.e., whether it's in (un) controlled flight. Development note: may be expressed using classes and properties for attitude state. | Datatype: Boolean |
| Has orbital period value | Indicates a specific value of the period of an object in orbit. | Datatype: Double<br>Example: |
| Has orbital velocity value | Indicates the specific numeric value of the velocity of an orbiting object. | Datatype: Double |
| Has launch date value | Indicates the specific date that a launch vehicle, artificial satellite, spacecraft, or payload is launched from a ground facility into orbit. | Datatype: xsd:date<br>Example: 1997-09-14 |
| Has launch site | Indicates the location of the launch, in string datatype. | Datatype: String<br>Example: Cape Canaveral, USA |
| Uses computational model | A generic relation linking a orbit dataset; orbit determination, propagation or conjunction assessment process; or estimate; to a particular model (e.g. force model). | |
| Has range value at time | Indicates the value, at a particular time, of the range property of a target object. | |
| Has radar cross section value | Alternative label: 'has RCS value'<br>Relates an object with the value of its RCS, indicated by a numerical value. | Datatype: Double |
| Has epoch day value | Indicates the specific day of the epoch in question. | |

Table 4. Data properties, domain and range, and data types (continued).

| Data Property Label | Description | Domain; Range/Datatype |
|---|---|---|
| Has data source | Links a space object observation, or a dataset to its source/origin, indicating the source with a string value. | Data object, information object, dataset<br>Datatype: String |
| Has source of orbit data | A specialized relation linking the orbit data (e.g. a metric dataset from a radar sensor) and where it came from, e.g., a particular ground-based sensor, or space agency).<br>Example: The source of a particular TLE may be a radar sensor of the U.S. Space Surveillance Network. | Orbit data, Orbit dataset, Observation, Observational dataset, TLE dataset, …<br>Space agency, Sensor device, SSA network, Sensor network<br>Datatype: String |

### 10.5.2 Object Properties

Table 5 presents a description of some OWL Object Properties. The domain and range are separated by a semicolon but included optionally. Multiple domain/range entities are separated by commas.

Table 5. Object properties with domains and ranges.

| Object Property Label | Description | Domain; Range |
|---|---|---|
| Was generated by | A generic relation linking an orbital debris object with an event.<br>Example formal statement: OrbitalDebrisFragment543 was_generated_by OrbitalCollisionEvent2001 | Orbital Debris Object , Event |
| Collided with | A relation between physical objects, e.g., artificial satellites. | |
| Describes | Relates an orbital data format or individual orbit dataset (e.g., two-line element set, orbital state vector) and a specific orbit at a particular time.<br>An asymmetric object property. | Orbit data format, Orbit Dataset; Gravitational orbit |
| Describes at time | A time-indexed relation between an orbit data format (e.g., TLE) and an orbit. | |
| Describes orbit of | Relates orbital datasets (e.g. a TLE) of an observed orbiting object with the object. | |
| Detected by | A relationship between an object and a sensor, whereby the latter detects the former. | |
| Detects | Links a sensor device, or space agency, with the detected object. | |
| Has epoch | Links an observation, or dataset, with a particular Epoch (or temporal/time reference frame).<br>An asymmetric object property. | Observation, Observation dataset, Orbit Dataset;<br>Epoch |
| Has expected lifetime | Links an artificial satellite with its planned operational lifetime, given in years. | Artificial satellite; Expected lifetime |
| Has function | A generic relationship linking the bearer of a function, and the function itself, e.g., between an engineered system and one of the functions of the system | |
| Has launch date | Links a spaceflight launch vehicle, spacecraft or other astronautical system to its date of launch from a ground facility. | Launch Vehicle, Spacecraft, …;<br>Launch date |
| Has launch operator | Links a launch vehicle and the operator of that vehicle. | |

Table 5. Object properties with domains and ranges (continued).

| Object Property Label | Description | Domain; Range |
|---|---|---|
| Has launch site | Links a launch vehicle, spacecraft or other astronautical system with the launch site. An asymmetric, functional object property. | |
| Has launch vehicle | Links a spacecraft other than the launch vehicle, such as a communications satellite payload, to its launch vehicle. | |
| Has mean anomaly | Links an orbit or orbiting object with its mean anomaly | Gravitational orbit; Mean anomaly |
| Has mission | Links an artificial satellite or spacecraft with its associated mission.<br>An asymmetric relation. | Artificial satellite, Spacecraft;<br>Mission,<br>Astronautical Mission, Spaceflight Mission |
| Has NORAD number | Links a space object and its assigned NORAD identifier number.<br>A functional object property. | Physical object, Space object, Orbital object, Resident space object; NORAD catalog number |
| Has operational status | Links an artificial satellite or non-orbiting spacecraft and its operational status. Clarifying note: The operational status may be those classes of status designation assigned by a governmental agency. | Artificial satellite, Spacecraft; Operational status |
| Has operator | A generic relation capable of linking an artificial satellite or spacecraft with its operator. | |
| Has orbit | Links an orbiting object with its orbit. | |
| Has orbital eccentricity | An orbit property relation linking an orbit, orbital trajectory or orbiting object, and its eccentricity. | Gravitational orbit; Orbital eccentricity |
| Has orbital inclination | An orbit property relation linking an orbit, orbital trajectory or orbiting object, and its inclination. | |
| Has orbital parameter | Alternative Label: 'has orbital property' | |
| Has orbital property | A generic relation linking an orbit or orbiting object and associated orbital properties/parameters (e.g. classical Keplerian orbital elements) or their values at a time. | |
| Has TLE set | A relation linking an orbit or orbiting object with a TLE at time.<br>Domain: Object, Orbital object, Space object, Resident space object, Spacecraft, Artificial satellite; Range: TLE | |
| Is observed by | A relation linking an observed object and the observer.<br>Domain: Physical_Object, Orbital_Object, Satellite, Spacecraft, Space object contact;<br>Range: Sensor_Device, Space agency, … | |
| Is operated by | Relates an astronautical system, spacecraft, payload, etc., and its operator.<br>Inverse of: operates | |
| Is orbited by | Links a central body and a satellite orbiting it.<br>The inverse property of the orbits property. | Central body, Central mass, Astronomical object, Primary body; Satellite, Orbital object |
| Is owned by | Links a satellite or spacecraft and its owner. | |
| Is part of | A generic mereological relation linking some entity (broadly construed) and another entity of which it is a part. Subrelation: 'is proper part of'. | |
| Is tracked by | Links an object (e.g. a satellite) or a detected object, and the device, facility, or organization tracking its position and motion over time.<br>Inverse of: tracks | |
| Observes | A generic relation linking an observer and the observed entity.<br>Example: An optical telescope observes an artificial satellite. | Sensor; Physical Object |
| Orbits | A generic relation between an orbiter (an orbiting object) and that which it orbits. It relates an entity and what it revolves around.<br>Development note: disambiguate from 'gravitationally orbits'. | |

Table 5. Object properties with domains and ranges (continued).

| Object Property Label | Description | Domain; Range |
|---|---|---|
| Has central body | A relation between an orbiting object and the mass it orbits.<br>Example formal expression: Earth has_Central_Body Sol | |
| Has country of origin | Indicates the country from which an artifact originated or the country that has authority over it. Links the spacecraft to it nation-state of origin. | |

### 10.6 Development Notes

The following holds in the version used for this TM and reflects common development tasks in the overall project. Not all definitions, axioms, and rules (e.g., SWRL) are present, but will be assigned, formatted, and added, pending further development. URIs/IRIs may not be active but may serve as temporary placeholders until finalized. IRI short names (for each class or property) are tentatively presented in natural language but may be in a universal or GUID form. Portions of the SSAO—selected classes and properties—will be grouped into modular ontologies, which can then be imported into (or mapped to) other knowledge organization systems such as thesauri, vocabularies, and other ontologies. For example, the orbital debris portion of the class hierarchy is from some of the Orbital Debris Ontology (ODO).[10] This is noted in appropriate places in the entity hierarchy with annotation properties for classes or object/data properties, specifically using the rdf:comment or dc:description metadata element or author-specified annotations, such as 'Development Status Note.' This ontology design reflects a modular architecture according to which the target domain is delineated into subdomains for more efficient computation, reuse, and knowledge management. Although presented in an OWL format serialization, various knowledge representation languages will be used to implement the ontology(s), affording greater expressivity and applicability. Contact the author with questions or interest in supporting development.

### 10.7 Future Development and Desiderata

Further development tasks include conceptual modeling; subject-matter analysis; class hierarchy structuring; scoping; ontology modularization of both generic and domain concepts; review, refinement, development and comparison of definitions; inference rules, axioms and constraints; implementation languages (OWL, CLIF, KIF, etc.), assigning permanent identifiers; data research; mappings; etc.

To fully develop the project vision in a sustainable manner, desiderata include: sponsors or formal opportunities, subject-matter experts, an interdisciplinary team, and technical resources/services, users, applications, and data sources. NASA and other space organizations are desired both as users and collaborative developers of the ontology(s). Collaboration with academia, industry, and government are also desired. Technical goals include, but are not limited to: a hosting platform (possibly web-based) with graphical-user interface, dynamic visualization of the ontology(s) in graph form, and dynamic search and retrieval.

The project envisions the development of a dynamic, computable, accurate, and potentially standardized knowledge model. As well as a metadata set, vocabulary, and ontology suite that can be applied to various applications (such as 3D visualizations, systems engineering, simulations, etc.) in order to support space situational awareness, spaceflight safety, and other applications. Please contact the author at rrovetto@terpalum.umd.edu if the reader is interested in using the ontology(s), contributing, collaborating, sponsoring, or supporting development.

## 11. CONCLUSION

Ontologies provide the capability to model a domain of knowledge through a semantically-rich and computable taxonomy of classes, relationships, properties, and annotations. Data import transformation rules enable the integration of datasets and generation of instantiated named individual objects with assigned values of specified data types to data properties. The compact JSON-LD format enables the development of webpages with tables and plots populated by data exported from an ontology.

This TM presented a demonstration of a virtual solar system that was generated from JSON-LD exported from the Space Situational Awareness Ontology (SSAO). Sections following the demonstration explained how to configure the Protégé ontology editor to receive imported data and an R tutorial that used classes from the SSAO to generate both a webpage with a table of orbital parameters and a dynamic 3D visualization of those parameters.

The SSA Ontology, while subject to revision, specifies a taxonomy of relevant astronautical concepts. It defines associated terms using data properties, axioms, computer- and human-readable definitions, and metadata annotations. Data integration and linked-data files are two benefits of the SSAO that were highlighted in this TM. This TM demonstrated how to apply and develop ontologies for web-based and visualization applications.

# 12. ABOUT THE AUTHORS

## 12.1 Daniel A. O'Neil

Employed at Marshall Space Flight Center since 1991, Dr. O'Neil has supported a variety of small and large projects in the roles of technical monitor, systems engineer, and project manager. He managed the development of the Virtual Research Center,[50] which was one of the first web-based intranets. In 1997, he coordinated a workshop on general public space travel and tourism.[51] In an agency-wide conceptual design and technology development project, the Space Solar Power Exploratory Research and Technology study,[52] he served as an assistant project manager, and supported the Constellation Program as a software system engineer. Presently, he works in the Office of Strategy where he analyzes technology trends, such as virtual and augmented reality, artificial intelligence, and ontology visualization.

Education:

- B.S. Electrical and Computer Engineering, 1985, University of Alabama in Huntsville.
- M.S. Engineering Management, 1997, University of Alabama in Huntsville.
- Ph.D. Modeling and Simulation, 2019, University of Alabama in Huntsville.

## 12.2 Robert J. Rovetto

Born in New York, Robert has been a member of the NASA Datanauts group (established by NASA's Chief Information Officer), since 2017, and a research affiliate of the Center for Orbital Debris Education and Research (University of Maryland) since 2016. He has an interdisciplinary background, which includes approximately 20 publications since 2011, ontology development, conceptual modeling, philosophy, and maritime operations. He has both independent and team-based experiences, as well as extensive volunteering. Presently, Robert is serving in semantic/knowledge technology committees, and astronautics-related committees, contributing in part to ontology and terminology-related tasks. Service-to-discipline includes for the International Association for Ontology and its Applications, the International Astronautical Federation, International Standards Organization, the American Institute of Aeronautics and Astronautics, the Earth Science Information Partners, and the Consultative Committee for Space Data Systems. Finally, he pursues training and service opportunities in maritime search and rescue. Areas of interest include knowledge representation and reasoning; formal and computational ontology; philosophy; model-based system engineering; astronautics; space situational awareness; space traffic management; space policy; AI; ethics; symbolic logic; and maritime safety.

Education:

- B.A. Philosophy, 2007, University of Maryland, College Park.
- M.A. Philosophy with ontology focus, 2011, State University of New York.
- Graduate coursework, Space Studies 2009, 2011, 2012, American Military University (APUS).
- Merchant Mariner Credential: Deck Officer, Mariner's School/Learning System, 2008, 2013, 2017.

## REFERENCES


1. Gruber, T.R.: "A translation approach to portable ontologies," *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199–220, 1993.

2. Raskin, R.; Pan, M.: "Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)," *Computers and Geosciences*, Vol. 31, No. 9, pp. 1119–1125, November 2005.

3. Sidney, S.C.; Hodgson, R.; and Keller, P.J.: "Large-Scale Knowledge Sharing for NASA Exploration Systems," NASA Ames Research Center, <https://franz.com/agraph/cresources/white_papers/NASA_Paper-2008.pdf>, 2008.

4. Keller, R.M.: "The NASA Air Traffic Management Ontology," NASA/TM—2017–219526, NASA Ames Research Center, Moffett Field, CA, 88 pp., June 2017.

5. "NASA Metadata Catalog," <https://data.nasa.gov/data.json>, Accessed 2020.

6. Rovetto, R.J.: "The Orbital Space Environment and Space Situational Awareness Domain Ontology – Towards an International Information System for Space Data," Paper Presented at The Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference, September 20–23, 2016.

7. Rovetto, R.J.: "Orbital Space Environment and Space Situational Awareness Domain Ontology," In CEUR Workshop proceedings for The Joint Ontology Workshops at the 9th International Conference of Formal Ontology for Information Systems (FOIS), Early Career Symposium, Annecy, France, July 2016.

8. Rovetto, R.J. and Kelso, T.S.: "Preliminaries of a Space Situational Awareness Ontology," Presented at 26th AIAA/AAS Space Flight Mechanics Meeting, Napa, CA, USA, February 14–18, 2016.

9. Rovetto, R.J.: "Ontology for Europe's Space Situational Awareness Program," Paper Presented at 7th European Conference on Space Debris, European Space Operations Centre Darmstadt, Germany, April 18–21, 2017.

10. Rovetto, R.J.: "An Ontological Architecture for Orbital Debris Data," *Earth Science Informatics*, Vol. 9, No. 1, pp. 67–82, doi:10.1007/s12145-015-0233-3, 2016.

11. Rovetto, R.J.: "The Orbital Space Domain Knowledge Modeling Project," <https://ontospace.wordpress.com/>, <https://purl.org/space-ontology>, Accessed September 2020.

12. Rovetto, R.J.: "An Ontology for Satellite Databases," *Earth Science Informatics*, Vol. 10, pp. 417–427, doi:10.1007/s12145-017-0290-x., April 2017.

13. Rovetto, R.J.; Kelso, T.S.; and O'Neil, D.A.: "Orbital Debris Ontology, Terminology, and Knowledge Modeling," *Journal of Space Safety Engineering*, Vol. 7, No. 3, pp. 451–458, September 2020.

14. Rovetto, R.J. "Ontology-based Knowledge Management for Space Data," in *68th International Astronautical Congress (IAC 2017): Unlocking Imagination, Fostering Innovation and Strengthening Security*, September 25–29, 2017, Curran Associates, Inc., Adelaide, Australia, 13 pp., 2017.

15. World Wide Web Consortium (W3C): "Web Ontology Language (OWL)," https://www.w3.org/OWL/, December 11, 2012.

16. Musen, M.A.; and the Protégé Team: "The Protégé Project: A Look Back and a Look Forward," *AI Matters*, Vol. 1, No. 4, pp. 4–12, doi:10.1145/2557001.25757003, June 2015.

17. Wikipedia: "Landsat 7," https://en.wikipedia.org/w/index.php?title=Landsat_7&oldid=826381859, May 22, 2018.

18. Zhang, Z.; Miller, J.A.: "Ontology Query Languages for the Semantic Web: A Performance Evaluation," <https://pdfs.semanticscholar.org/b7ac/ba0651f16859a2eb92eee2be52895b1541f2.pdf>, 2015.

19. World Wide Web Consortium (W3C): "Resource Description Framework (RDF)," <https://www.w3.org/2001/sw/wiki/RDF>, February 25, 2014.

20. International Organization for Standardization: "Information technology—Common Logic (CL): a framework for a family of logic-based languages," ISO/IEC 24704:2007, Vernier, Geneva, Switzerland, 73 pp., 2007.

21. World Wide Web Consortium (W3C): "SPARQL Query Language," <https://www.w3.org/TR/rdf-sparql-query/>, January 15, 2008.

22. Harrocks, I.; Patel-Schneider, P.F.; Boley, H.; et al.: "Semantic Web Rule Language (SWRL)," World Wide Web Consortium (W3C), <https://www.w3.org/Submission/SWRL/>, May 21, 2004.

23. Pagels, M.: "The DARPA Agent Markup Language," DAML, <http://www.daml.org/>, January 29, 2006.

24. TopQuadrant: "TopBraid Composer – Maestro Edition," <https://www.topquadrant.com/tools/IDE-topbraid-composer-maestro-edition/>, Accessed September 2020.

25. Semafora: “OntoBroker and OntoStudio X,” <https://www.semafora-systems.com/ontobroker-and-ontostudio-x>, 2017.

26. Knowledge Media Institute: “Apollo,” <http://kmi.open.ac.uk/technologies/name/apollo>, June 22, 2005.

27. Kozaki, K.: “Hozo Ontology Editor,” Institute of Scientific and Industrial Research, Osaka University, <http://www.hozo.jp/>, February 16, 2018.

28. Institute of Mathematics and Computer Science, University of Latvia: “OWLGred,” <http://owlgred.lumii.lv/>, Accessed September 2020.

29. Cognitum: “Fluent Editor for PC,” <http://www.cognitum.eu/Semantics/FluentEditor/>, Accessed September 2020.

30. Rovetto, R.J.: “rrovetto,” Github, <https://github.com/rrovetto>, Accessed September 2020.

31. Alatrish, E.: “Comparison Some of Ontology Editors,” *Management Information Systems*, Vol. 8, No. 2, pp. 18–24, 2013.

32. Hozo: “Ontology Editor (Distributed Development Version) Operating Manual, The Institute of Scientific and Industrial Research, Osaka University,” <hozo.jp/HozoManual_en_20140317.pdf>, September 2011.

33. W3C Recommendation: “JSON-LD 1.0, A JSON-based Serialization for Linked Data,” <https://www.w3.org/TR/json-ld/>, January 16, 2014.

34. ECMA International: “The JSON Data Interchange Format,” <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>, October 2013·

35. W3Schools.com: “JSON.Parse(),” https://www.w3schools.com/js/js_json_parse.asp.

36. Rovetto, R.J.: “The Space Situational Awareness Ontology,” <https://github.com/rrovetto/space-situational-awareness-domain-ontology/> and < https://purl.org/space-ontology/ssao>, 2011.

37. Rovetto, R.J.: “The Orbital Debris Ontology,” <https://purl.org/space-ontology/odo>, 2011.

38. PoolParty Semantic Suite: “Ontology Management,” <https://www.poolparty.biz/ontology-management/>, Accessed September 2020.

39. JSON-LD: “JSON for Linking Data,” <http://json-ld.org>, Accessed September 2020.

40. Wikipedia: “Orrery,” <https://en.wikipedia.org/w/index.php?title=Orrery&oldid=766914621> Accessed March 10, 2017.

41. Digital Bazaar GitHub: "A JSON-LD Processor and API implementation in JavaScript," https://github.com/digitalbazaar/jsonld.js, Accessed September 2020.

42. Logsdon, T.: *Orbital Mechanics: Theory and Applications*, John Wiley & Sons, New York, NY, 288 pp., 1998.

43. "Protégé 5 Documentation," Stanford Center for Biomedical Informatics Research, Stanford University School of Medicine, <http://protegeproject.github.io/protege>

44. Ooms, O.: "The jsonld Package: JSON for Linking Data," https://cran.r-project.org/web/packages/jsonld/jsonld.pdf>, April 11, 2017.

45. Ooms, J.: "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects," <https://arxiv.org/abs/1403.2805>, 2014.

46. Adler, D.; and Murdoch, D.: "The rgl Package: 3D Visualization using OpenGL," <https://cran.r-project.org/web/packages/rgl/rgl.pdf>, March 28, 2018.

47. Rovetto, R.J.: "The Orbital Space Domain Knowledge Modeling Project," <http://rrovetto.github.io/Orbital-Space-Ontology-Project/>, Accessed September 2020.

48. NASA: Orbital Elements, <https://spaceflight.nasa.gov/realdata/elements/>, Accessed September 2020.

49. Hejduk, M.D.; and Frigm, R.C.: "Collision Avoidance Short Course: Conjunction Assessment Risk Analysis - NASA Robotic Cara," Paper Presented at Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Maui, HI, September 15–18, 2015.

50. Mankins, J.C.; and O'Neil, D.A.: "The Virtual Research Center: A new paradigm for collaboration among geographically distributed teams," Paper Presented at Space Programs and Technologies Conference, Huntsville, AL, September 26–28, 1995.

51. O'Neil, D.; Bekey, I.; Rodgers, T.; and Stallmer, E.: "General Public Space Travel and Tourism Volume 1 Executive Summary," NASA/NP-1998-03-11-MSFC/VOL1, <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19990032557.pdf>, March 1998.

52. Mankins, J.C.; Howell, J.T.; and O'Neil, D.A.: "New Concepts and Technologies from NASA's Space Solar Power Exploratory Research and Technology Program", <https://space.nss.org/wp-content/uploads/Space-Manufacturing-conference-13-077-NASA-SERT-Program.pdf>, 2001.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 01–01–2021 | Technical Memorandum | |

**4. TITLE AND SUBTITLE**

An Ontology-Based Virtual Orrery

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

D.A. O'Neil, R.J. Rovetto*

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

George C. Marshall Space Flight Center
Huntsville, AL 35812

**8. PERFORMING ORGANIZATION REPORT NUMBER**

M–1516

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

National Aeronautics and Space Administration
Washington, DC 20546–0001

**10. SPONSORING/MONITOR'S ACRONYM(S)**

NASA

**11. SPONSORING/MONITORING REPORT NUMBER**

NASA/TM–20210000030

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Unclassified-Unlimited
Subject Category 61
Availability: NASA STI Information Desk (757–864–9658)

**13. SUPPLEMENTARY NOTES**

Prepared by the Office of Strategic Analysis and Communications
*NASA Datanauts, New York, New York

**14. ABSTRACT**

Tutorials in this technical memorandum explain how to use the Protégé ontology editor to import data from Excel and export data from Protégé in the JavaScript Object Notation Linked Data (JSON-LD) format then how to transform the JSON-LD file into variables for web apps. These tutorials apply JavaScript and R programming languages and 3D graphics libraries. Ontologies enable standardization, data sharing, and semantic interoperability. An ontology models a domain of knowledge as taxonomies of annotated classes, data properties, relational object properties, and inference rules. This technical memorandum includes a data dictionary for a subset of the Space Situational Awareness Ontology (SSAO).

**15. SUBJECT TERMS**

Ontologies, Web-apps, 3D graphics, Orbits, Trajectories, Visualization

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | STI Help Desk at email: help@sti.nasa.gov |
| U | U | U | UU | 88 | 19b. TELEPHONE NUMBER *(Include area code)* STI Help Desk at: 757–864–9658 |

National Aeronautics and
Space Administration
IS02
**George C. Marshall Space Flight Center**
Huntsville, Alabama 35812