Ontologies (in a nutshell)

What are they?

In philosophy, Ontology is the “science of what is”.

In information science, an ontology is a formal (machine-readable) representation of knowledge within a certain domain. It identifies the categories (“classes”) that exist in the domain and the relations between them.

Why are they useful?

Ontologies allow 1) automatic reasoning, 2) easier exchange of information across heterogeneous sources.

What is the bigger picture?

The context is that of semantic technologies and semantic interoperability. Notably, the Semantic Web concept, an evolution of the World Wide Web that is based on semantics rather than ad-hoc links between resources (e.g., web-pages) was proposed in the 1990s.
Taxonomies

A taxonomy is a hierarchical classification.

Simpler than an ontology, has only sub-class relations (also called “is-a” relation). For example, think of taxonomies in botanics and zoology.

In most cases, its graph structure is an acyclic tree (Aristotelic approach). However, some authors allow multiple inheritance (i.e., a class can have multiple super-classes).

Metadata

Metadata are data that provide information about other data.

For example, given a text (data) one can annotate it with a title, author, language used etc (metadata). Given a software (data) one can annotate it with a version number, programming language etc (metadata).

In general, the concepts used as metadata can be just a list, or be organized in some structure. In our case, we use concepts belonging to ontologies and taxonomies.
The semantic web

**Triples:** Individual Relation Individual. (Subject Predicate Object.)

**Example**

(1) Frank is_father_of Robert.

Human a Class.

Frank a Human.

Robert a Human.

**Cardinality restriction:**

Every human has exactly one father.

**Principle: Open world assumption**

Since relevant information may distributed over the entire semantic web, rather than the presently considered source only, the available knowledge is assumed to be incomplete.

(Contrast with a closed, monolothic database architecture.)
The semantic web

**Triples: Individual Relation Individual.** (Subject Predicate Object.)

Example

(1) Frank is_father_of Robert.

Q: “Is Emanuele the father of Robert?”

**Principle: Open world assumption**
Since relevant information may distributed over the entire semantic web, rather than the presently considered source only, the available knowledge is assumed to be incomplete.

(Contrast with a closed, monolithic database architecture.)

**Principle: No unique name assumption**
Unless explicitly stated, individuals with different names can coincide.
The semantic web

Example

(1) Frank is_father_of Robert.

Q: “Is Emanuele the father of Robert?”
A: “We don’t know.”

Principle: Open world assumption
Since relevant information may distributed over the entire semantic web, rather than the presently considered source only, the available knowledge is assumed to be incomplete.
(Contrast with a closed, monolithic database architecture.)

Principle: No unique name assumption
Unless explicitly stated, individuals with different names can coincide.
The semantic web

Triples: Individual Relation Individual. (Subject Predicate Object.)

Example

(1) Frank is_father_of Robert.             Human a Class.

(2) Frank is_different_from Emanuele.    Frank a Human.
                                             Robert a Human.

Q: “Is Emanuele the father of Robert?”

A: “No, he is not.”

Principle: Open world assumption
Since relevant information may distributed over the entire semantic web, rather than the presently considered source only, the available knowledge is assumed to be incomplete.
(Contrast with a closed, monolithic database architecture.)

Principle: No unique name assumption
Unless explicitly stated, individuals with different names can coincide.
The semantic web

Example

(1) Frank is_father_of Robert.

(2) Frank is_different_from Emanuele.

Q: “Is Emanuele the father of Robert?”

A: “No, he is not.”

(3) Frank is_father_of Anna.

Q: “How many children does Frank have?”

A: “How many different entities X are there such that Frank is_father_of X?”

Triples: Individual Relation Individual. (Subject Predicate Object.)
The semantic web

Example

(1) Frank is_father_of Robert.  
(2) Frank is_different_from Emanuele.  
(3) Frank is_father_of Anna.

Q: “Is Emanuele the father of Robert?”  
A: “No, he is not.”

Q: “How many children does Frank have?”  
A: “At least two.”

Open world assumption: More information could be available elsewhere.
Ontologies on the marketplace

How do we use them in a marketplace?

They cover all aspects of the marketplace. At the user interface, for the data ingest (i.e., to create entries about agents, products, services) and to guide search and browsing. Internally, they are the base of interoperability of the marketplace components.

How will this help/affect users and providers?

Ontologies define the “common language” used in the marketplace. Users will indirectly see them via the available keywords and search criteria and results (e.g., as when you search a movie on Netflix).

This vocabulary will not be frozen: there will be a policy to allow users/providers to request the addition of categories or to integrate sub-domain specific ontologies.

Providers can choose down to which level of detail to adhere to the proposed common language: the deeper the adherence, the deeper the interoperability with other services.
Ontology development on the basis of the EMMO

Types of relations covered by the European Materials & Modelling Ontology (EMMO)

1) **Taxonomy**: Subclass relation (between classes)

2) **Semiosis**: Representation of *physical* entities by *signs*

3) **Mereotopology**: Parthood (of a part in a *fusion*) and slicing

4) **Set theory**: Membership (of an element in a *set*)

![Diagram of sign, object, and interpretant]

![Classification of entities: electronic, atomistic, mesoscopic, continuum]

C. S. Peirce
Ontology development on the basis of the EMMO

Types of relations

1) **Taxonomy**: Subclass relation
2) **Semiosis**: Representation by *signs*
3) **Mereotopology**: Parthood and slicing
4) **Set theory**: Membership

Branches and important classes from EMMO

1) “material”
2) “process”
3) “quantitative property”
4) “model”
5) “qualitative property”
6) “semiotic”
Ontology development on the basis of the EMMO

MODA Graph Language, CEN Workshop Agreement 17284, and EMMO (Ghedini et al.)

- European Virtual Marketplace Ontology (EVMPO)
  - Persons, institutions, stakeholders (CVIII)
  - Computational resources (MACRO)
  - Models and simulation workflows (OSMO)
  - Training services and competencies (OTRAS)
  - Communication and messages (VICO)
  - Validation, benchmarking, assessments (VIVO)
  - Translation, decision support (MMTO)
  - Software, quantities, variables (VISo + branches + VOV)

- Upper level: EMMO extended by European Virtual Marketplace Ontology (EVMPO)
- Marketplace-level ontologies: VIMMP in coordination with the MarketPlace project
- Subdomains: VOV, VISo branches (electronic, atomistic-mesoscopic, continuum)
European Virtual Marketplace Ontology

The EVMPO provides a structure for the marketplace-level ontologies by formulating fundamental paradigmatic categories that correspond to irreducible terms which are seen as constitutive to the virtual-marketplace paradigm.

**Recommendation:** Any ontology at the marketplace level should follow the structure given by these categories as closely as possible.

**Fundamental paradigmatic categories:**

1. *assessment*, i.e., proposition on accuracy, performance of an entity, or of an entity’s trust in another entity
2. *calendar_event*, i.e., meeting or activity that is scheduled or can be scheduled, equivalent to Vevent from ICALTZD
3. *communication*, i.e., statement or sequence of statements that can be communicated at a virtual marketplace
4. *information_content_entity* as defined in the Information Artifact Ontology (IAO)
5. *infrastructure*, i.e., virtual-marketplace infrastructure (e.g., data access, hardware, and software)
6. *material* as defined in the European Materials Modelling Ontology (EMMO)
7. *model*, i.e., entity that can be described by the 2nd section of MODA, equivalent to "model" from the EMMO
8. *process*, i.e., temporal evolution of one or multiple entities
   *note:* *physical_process* is a subclass of EMMO "process"
9. *product*, i.e., good or service that can be offered either at a virtual marketplace or off-site
10. *property* as defined in the EMMO
11. *role* as defined in the EMMO
12. *simulation*, i.e., a simulation workflow (as in MODA)
Landscape of semantic assets developed by VIMMP

EMMC line of semantic asset development:

1) Review of Materials Modelling (RoMM)
2) CWA 17284 – Model Data (MODA)
3) European Materials & Modelling Ontology (EMMO)

Blue: Semantic assets co-developed by VIMMP
VISO – VImmp Software Ontology

VISO’s main purpose is to describe software, addressing mostly its capabilities (both model and solver aspects), but also licensing, requirements (as libraries and operating systems) and compatibility\(^1\) with other tools.

It will be used to structure the ingestion of information about software tools on the virtual marketplace. The same keywords will be then available to the users to browse the tools and compare them.

[1] Following E. Ghedini (EMMC), we distinguish between compatibility and interoperability, namely:

**compatibility** (=ability to exchange information directly, no need to interface)

**interoperability** (=ability to exchange information through a common language)

Categories at the upper level:

(1) **agent** = An entity (individual, group, institution) that can potentially act on a virtual marketplace

(2) **software** = A computer program. Can be a software tool, a compiler, or an operating system.

(3) **license** = Regulation of the right to use, modify and distribute something, in this case software.

(4) **programming_language** = A language that can be used to write software.

(5) **solver_feature** = Capability of a software tool, intended as a numerical algorithm which is implemented.

(6) **model_feature** = Capability of a software tool, intended as a model aspect that can be addressed.

(7) **modelling_related_entity** = High level concept related to modelling, as statistical mechanics, the RoMM models, fundamental physics equation, etc.

(8) **property** = A feature that can be measured or computed

(9) **software_update** = Allows to describe the differences between two softwares. It has input/output the older/newer version of the software

(10) **software_interface** = Interface between a software and a user or a client (i.e., a program or device)
VISO (2): The structure

Below an upper level (viso-general) that addresses aspects common to all software, we split VISO into three branches, i.e., electronic (EL, viso-el), atomistic-mesoscopic (AM, viso-am) for the two molecular granularity levels from RoMM, and continuum (CO, viso-co).

The model_feature class has typically a rich structure, so we further subdivide it into three classes. As an example, we show here the am_model_feature class and its subclasses.

VISO was designed having in mind Molecular Dynamics (MD), Dissipative Particle Dynamics (DPD), Quantum Density Functional Theory (DFT) and Computational Fluid Dynamics (CFD) models and tools.
VISO (3): Solver features

Here, we show instead the `solver_feature` class and its subclasses, for the three branches.

This includes, for example, concepts as: `geometric_constraint_algorithm`, `electrostatics_solver`, `integrator` (in AM), `ionic_relaxation`, `basis_set` (in EL), `spatial_discretization_scheme` (in CO).

Main contributors of VISO:

Silvia Chiacchiera, Martin T. Horsch
Mara C. Chiricotto, Joshua D. Elliott (UoM)
Gianluca Boccardo (Polito)
The main relations between objects in VISO are:

**has_feature** = To describe the features of a software tool [Inverse: **is_feature_of**]

**is_tool_for_model** = Relates software tools and RoMM models

**is_compatible_with** = Asserts compatibility between software tools

**is_distributed_by** = Relates tools and agents [Inverse: **is_distributor_of**]

**has_license** = Relates software and license

**requires** = Relates a software tool to libraries and/or operating systems

**can_run_on** = Relates a software tool to operating systems

Some relations between objects and literals are:

**is_free, is_open_source, is_a_library, has_a_gui**

Example for a software tool (extract from a .TTL file):

```
ex:DL_POLY a viso:software_tool;
viso:is_free false;
viso:is_free_to_academic true;
viso:has_a_gui true;
viso:is_open_source true;
viso:is_a_library false;
viso:is_distributed_by ex:STFC;
rdfs:seeAlso "https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx"^^xs:anyURI;
viso:is_tool_for_model viso-am:MM;
viso:is_tool_for_model viso-am:MD;
viso:has_feature viso-am:DOMAIN_DECOMPOSITION;
viso:has_feature viso-am:DIRECT_COULOMB_SUM;
viso:has_feature viso-am:SPME;
viso:uses_language viso:FORTRAN90;
viso:has_feature viso-am:NVE;
viso:has_feature viso-am:NVT;
viso:has_feature viso-am:NPT;
viso:has_feature viso-am:LENNARD_JONES_12_6;
viso:has_feature viso-am:RIGID_BOND;
viso:has_feature viso-am:VV;
viso:has_feature viso-am:LFV;
viso:is_compatible_with ex:PLUMED;
viso:is_compatible_with ex:OPENKIM;
a viso:simulation_engine.
```
VOV – Vimmp Ontology of Variables

The purpose of VOV is to **organize the variables** (in broad sense, including constants) that appear in modeling and simulations, and to connect them to models and algorithms in which they are involved and to model objects (e.g., entities entering a simuations, such as sites, rigid bodies) **to which they are attached**.

VOV is to be used in connection to VISO and OSMO, to further specify models, algorithms and workflows.

See the examples on the right.
The OTRAS ontology includes topic and operator catalogues.

- **mm_topic_basic** (codes 1XXX and 2XXX): Basic prerequisites for materials modelling.
- **mm_topic_computational** (codes 3XXX): Computational and numerical aspects of materials modelling.
- **mm_topic_data** (codes 4XXX): Data science and technology aspects.
- **mm_topic_materials** (codes 5XXX): Topics related to fluid and solid materials.
- **mm_topic_social** (codes 6XXX): Social, economic, and community aspects.
- **mm_topic_theoretical** (codes 7XXX): Theory (non-computational aspects).
- **mm_topic_interdisciplinary** (codes 8XXX)
- **mm_topic_side** (codes 9XXX): Topics from other disciplines

Pre-existing semantic assets considered for further development:

- ACM Computing Classification System (CCS)
- APS Physics Subject Headings (PhysH)
Ontology use during data ingest

Expert competency description: “The expert \( X_1 \) can (accomplish) \( X_2 \) with respect to topic \( X_3 \) by doing \( X_4 \); for example, \( X_5 \).” (Note: \( X_4 \) and \( X_5 \) are not required.)

\[ \text{mm_topic_basic (codes 1XXX and 2XXX):} \]
Basic prerequisites for materials modelling.

\[ \text{mm_topic_computational (codes 3XXX):} \]
Computational and numerical aspects of materials modelling.

\[ \text{mm_topic_data (codes 4XXX):} \]
Data science and technology aspects.

\[ \text{mm_topic_materials (codes 5XXX):} \]
Topics related to fluid and solid materials.

\[ \text{mm_topic_social (codes 6XXX):} \]
Social, economic, and community aspects.

\[ \text{mm_topic_theoretical (codes 7XXX):} \]
Theory (non-computational aspects).

\[ \text{mm_topic_interdisciplinary (codes 8XXX):} \]

\[ \text{mm_topic_side (codes 9XXX):} \]
Topics from other disciplines

\[ \text{6120 chemical} \]
\[ \text{6140 automotive, aerospace, etc.} \]
\[ \text{6150 biotechnology} \]
\[ \text{6155 food} \]
\[ \text{6160 medicine} \]
\[ \text{6165 paper} \]
\[ \text{6170 electrical} \]
\[ \text{6175 machinery} \]
\[ \text{6180 metal (basic and fabricated)} \]
\[ \text{6190 special topics} \]

Ontology use during data ingest

Expert competency description: “The expert $X_1$ can (accomplish) $X_2$ with respect to topic $X_3$ by doing $X_4$; for example, $X_5$. (Note: $X_4$ and $X_5$ are not required.)


mm_topic_basic (codes 1XXX and 2XXX): Basic prerequisites for materials modelling.

mm_topic_computational (codes 3XXX): Computational and numerical aspects of materials modelling.

mm_topic_data (codes 4XXX): Data science and technology aspects.

mm_topic_materials (codes 5XXX): Topics related to fluid and solid materials.

mm_topic_social (codes 6XXX): Social, economic, and community aspects.

mm_topic_theoretical (codes 7XXX): Theory (non-computational aspects).

mm_topic_interdisciplinary (codes 8XXX)

mm_topic_side (codes 9XXX): Topics from other disciplines

6120 chemical
6140 automotive, aerospace, etc.
6150 biotechnology
6155 food
6160 medicine
6165 paper
6170 electrical
6175 machinery
6180 metal (basic and fabricated)
6190 special topics
Ontology use within the virtual marketplace framework

Virtual-marketplace stakeholder communication is formalized by VICO, taking into account specific requirements related to assessment and validation (VIVO) and translation.
# VIMMP Validation Ontology (VIVO) Assessment Matrix

<table>
<thead>
<tr>
<th>Absolute</th>
<th>Relative</th>
<th>Qualitative</th>
<th>(computing) time</th>
<th>Space (memory)</th>
<th>Other</th>
<th>Endorsement</th>
<th>Comment</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Requirement</td>
<td>Review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Agent</th>
<th>Data Item</th>
<th>Document</th>
<th>Event</th>
<th>Data</th>
<th>Hardware</th>
<th>Infrastructure</th>
<th>Software</th>
<th>Meta-Assessment</th>
<th>Model</th>
<th>Project</th>
<th>Data Access</th>
<th>Hardware Access</th>
<th>Software Access</th>
<th>Service</th>
<th>Training</th>
<th>Translation</th>
<th>Other</th>
<th>Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

*The matrix compares the evaluation criteria for different elements of the VIMMP validation ontology, with '+' indicating a positive match and '-' indicating no match.*
Ontology use within the virtual marketplace framework

self-descriptions: STFC_DARESBURY_LABORATORY

a cviii:institution, cviii:model_provider, cviii:software_owner, cviii:training_provider, cviii:translator;
cviii:has_name "STFC Daresbury Laboratory"^^xs:string;
cviii:has_interest

otras:MM_TOPIC_3000, otras:MM_TOPIC_6120, otras:MM_TOPIC_7200, otras:MM_TOPIC_8350;
cviii:has_interlocutor_tag [ 

cviii:is_academic true;
cviii:is_based_in lcc-codes:UnitedKingdomOfGreatBritainAndNorthernIreland;
cviii:is_in_group 

cviii:IG_MODEL_PROVIDER, cviii:IG_SOFTWARE_OWNER,
cviii:IG_TRAINING_PROVIDER, cviii:IG_TRANSLATOR;
cviii:is_for_profit false;
cviii:is_nuclear true;
cviii:is_sme false 
].
Ontology use within the virtual marketplace framework

VIMMP
VIRTUAL MATERIALS MARKETPLACE

CVIII
Categorization of VIMMP Institutions and Individuals

emmo:role

model provider

end user
translator
Semantic and pragmatic interoperability

end user

problem statement (w. OSMO use case)

interest statement

reformulated problem statement
(w. OSMO use case)

translator

reformulated solution statement (w. OSMO simulation workflow)

solution statement (w. OSMO simulation workflow)

model provider

acceptance statement, project opening statement, etc.
Semantic and pragmatic interoperability

Translation ontology

MODA & OSMO

problem statement
solution statement
Interoperability and data provenance

Decision Support

Repository

Translation Environment

Simulation Platform

user

Ingest
Search Download

Ingest
Search Download

Ingest
Search Download

Ingest
Search Download

Ingest
Search Download

Ingest
Search Download

Ingest
Search Download

Bespoke Functionality
(e.g., “I would like this as an input file for code X”)
Ontology use for data provenance characterization

MODA section 1

use_case

has_use_case_aspect

has_use_case_description,
has_use_case_material,
has_use_case_timespan,
has_use_case_geometry,
has_use_case_boundary_condition,
has_use_case_literature

MODA section 2

model

MODA section 3

solver

MODA section 4

processor

OSMO

“sections”

“aspects”

“graphs”

1. Aspect of the User Case/System to be simulated

1.1 ASPECT OF THE USER CASE TO BE SIMULATED

Describe the aspects of the user case textually.

No modelling information should appear in this box. This case could also be simulated by other models in a benchmarking operation.

The information in this chapter can be end-user information, measured data, library data etc. It will appear in the pick cycle of your workflow picture.

Simulated input which would have been calculated by another model should not be included (as in chapter 2.4).

Also the result of pre-processing necessary to translate the user case specifications to values for the physics variables of the entities can be documented here.

1.2 MATERIAL

Describe the chemical composition – and the values used for properties and from which database these are taken. If pre-processing was needed please specify the methodology.

1.3 GEOMETRY

Size, form, picture of the system (if applicable)

Note that computational choices like simulation boxes are to be documented in chapter 3.

1.4 TIME LAPSE

Duration of the case to be simulated.

This is the duration of the simulation. This is not the same as the computational forces to be done in chapter 3.

1.5 MANUFACTURING PROCESS OR IN-SERVICE CONDITIONS

E.g. heated walls, external pressures and bending forces.

Please note that these might appear as terms in the FE etc as boundary conditions and this will be documented in the relevant chapters.

Note: These conditions will be expressed in physics relations in Chapter 2.4.

Please specify the values used for parameters and from which database these are taken. If pre-processing was needed please specify the methodology.

1.6 PUBLICATION ON THIS DATA

Publication documenting the simulation with this single model (if available) and if not already included in the overall publication.
Ontology use for data provenance characterization

http://www.vimmp.eu/semantics/osmo/osmo.ttl

arXiv:1908.02335 [cs.DC]
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 760907.

This document and all information contained herein is the sole property of the VIMMP Consortium (unless specified otherwise or clear by context). Information presented herein may be subject to intellectual property rights. No intellectual property rights are granted by the delivery of this document or the disclosure of its content. Reproduction or circulation of this document to any third party is prohibited without the consent of the authors.

The statements made herein do not necessarily have the consent or agreement of the VIMMP Consortium. They represent the opinion and findings of the authors.

©2019 all rights reserved.