

GeoVocamp DC 2014: Computational Observation Ontology Design Pattern

Charles F. Vardeman II
University of Notre Dame
12/01/14

Linked Open Data for Computational Science

An Example from Computational Chemistry



Credit: Beatrice Murch, Creative Commons License, <https://www.flickr.com/photos/blmurch/2754681293/sizes/l/>



Physics > Fluid Dynamics

Why do bubbles in Guinness sink?

E. S. Benilov, C. P. Cummins, W. T. Lee

(Submitted on 23 May 2012)

Stout beers show the counter-intuitive phenomena of sinking bubbles while the beer is settling. Previous research suggests that this phenomena is due the small size of the bubbles in these beers and the presence of a circulatory current, directed downwards near the side of the wall and upwards in the interior of the glass. The mechanism by which such a circulation is established and the conditions under which it will occur has not been clarified. In this paper, we demonstrate using simulations and experiment that the flow in a glass of stout depends on the shape of the glass. If it narrows downwards (as the traditional stout glass, the pint, does), the flow is directed downwards near the wall and upwards in the interior and sinking bubbles will be observed. If the container widens downwards, the flow is opposite to that described above and only rising bubbles will be seen.

Comments: 5 pages, 4 figures. The movie referred to in the text is available as an ancillary file

Subjects: Fluid Dynamics (physics.flu-dyn); Popular Physics (physics.pop-ph)

Cite as: arXiv:1205.5233 [physics.flu-dyn]

(or arXiv:1205.5233v1 [physics.flu-dyn] for this version)

Download:

- PDF
- PostScript
- Other formats

Ancillary files (details):

- tilted_cylinder.avi

Current browse context:

physics.flu-dyn

< prev | next >

new | recent | 1205

Change to browse by:

physics

physics.pop-ph

References & Citations

- NASA ADS

2 blog links (what is this?)

Bookmark (what is this?)





About 685,000 results (0.16 seconds)

Water model - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Water_model ▾ Wikipedia ▾

The potential for models such as TIP3P and **TIP4P** is represented by. $E_{\{ab\}} = \sum_i k_C \text{exp}(-r_{OH}/r_{HOH})$, where k_C , the electrostatic constant, has a value of 332.1 ...

[Simple water models](#) - 2-site - 3-site - 4-site

TIP4P model of water page on SklogWiki - a wiki for ...

www.sklogwiki.org/SklogWiki/index.php/TIP4P_model_of_water ▾

Jan 20, 2011 - The **TIP4P** model is a rigid planar four-site interaction potential for water, ... The **TIP4P** model consists of a Lennard-Jones site for the oxygen ...

[Parameters](#) - [Phase diagram](#) - [Shear viscosity](#) - [Virial coefficients](#)

Water models

www.lsbu.ac.uk/water/models.html ▾ London South Bank University ▾

Apr 1, 2014 - Water molecular models including SPC, SPC/E, TIP3P, **TIP4P**, TIP5P, PPC, POL5, SSD and SWFLEX.

pair_style lj/cut/coul/long - Lammps

lammps.sandia.gov/doc/pair_lj.html ▾ Sandia National Laboratories ▾

style = lj/cut or lj/cut/coul/cut or lj/cut/coul/debye or lj/cut/coul/dsf or lj/cut/coul/long or lj/cut/coul/msm or lj/cut/**tip4p**/long; args = list of arguments for a particular ...

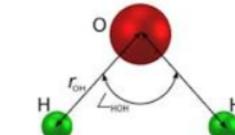
[PDF] TIP4P-Ew - Stanford University

www.stanford.edu/.../horn_tip4pEW_2004jcp.pdf ▾ Stanford University ▾

by HW Horn - 2004 - Cited by 557 - Related articles

May 22, 2004 - A re-parameterization of the standard **TIP4P** water model for use with Ewald techniques is introduced, providing an overall global improvement ...

Water model



In computational chemistry, classical water models are used for the simulation of water clusters, liquid water, and aqueous solutions with explicit solvent. These models use the approximations of molecular mechanics. [Wikipedia](#)

Related topics

In most water models, the **Lennard-Jones** term applies only to the interaction between the oxygen atoms. [Wikipedia](#)
Explore: [Lennard-Jones potential](#)

In-silico (see: water models), cyclic **water clusters** . . . are found with n = 3 to 60.

[Wikipedia](#)

Explore: [Water cluster](#)

Feedback

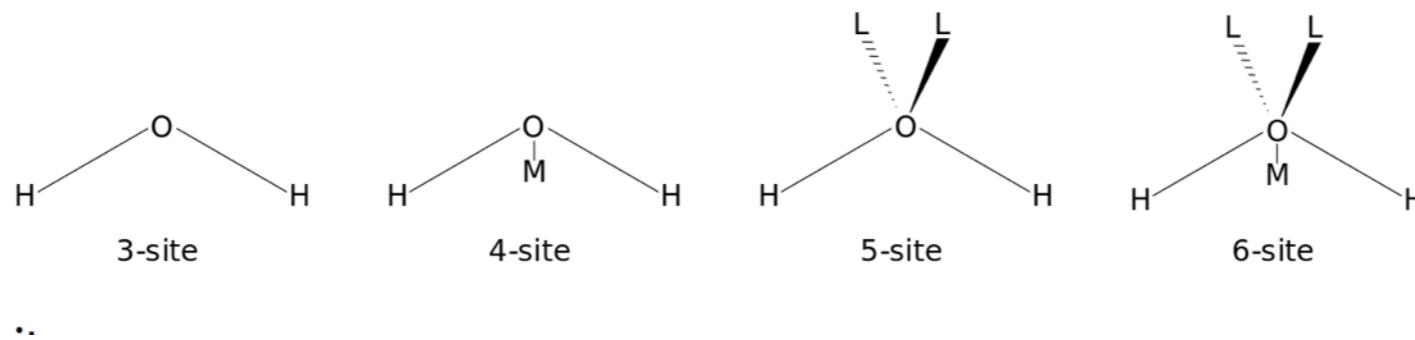
Simple water models [\[edit\]](#)

The simplest water models treat the water molecule as rigid and rely only on [non-bonded interactions](#). The electrostatic interaction is modeled using [Coulomb's law](#) and the dispersion and repulsion forces using the [Lennard-Jones potential](#).^{[1][2]} The potential for models such as TIP3P and TIP4P is represented by

$$E_{ab} = \sum_i^{\text{on } a} \sum_j^{\text{on } b} \frac{k_C q_i q_j}{r_{ij}} + \frac{A}{r_{\text{OO}}^{12}} - \frac{B}{r_{\text{OO}}^6}$$

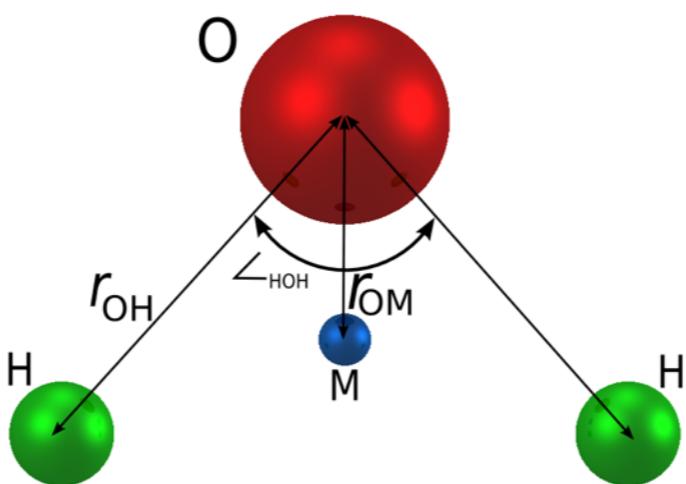
where k_C , the [electrostatic constant](#), has a value of $332.1 \text{ \AA}\cdot\text{kcal/mol}$ in the units commonly used in molecular modeling [[citation needed](#)]; q_i are the [partial charges](#) relative to the charge of the electron; r_{ij} is the distance between two atoms or charged sites; and A and B are the Lennard-Jones parameters. The charged sites may be on the atoms or on dummy sites (such as lone pairs). In most water models, the Lennard-Jones term applies only to the interaction between the oxygen atoms.

The figure below shows the general shape of the 3- to 6-site water models. The exact geometric parameters (the OH distance and the HOH angle) vary depending on the model.



Parameters [\[edit\]](#)

The TIP4P model consists of a [Lennard-Jones site](#) for the [oxygen](#) atom, and three charge sites.



r_{OH} (Å)	\angle_{HOH} , deg	σ (Å)	e/k (K)	$q(\text{O})$ (e)	$q(\text{H})$ (e)	$q(\text{M})$ (e)	r_{OM} (Å)
0.9572	104.52	3.154	78.0	0	0.52	-2q(H)	0.15

Velocity Verlet [edit]

A related, and more commonly used, algorithm is the **Velocity Verlet** algorithm,^[5] similar to the [leapfrog method](#), except that the velocity and position are calculated at the same value of the time variable (Leapfrog does not, as the name suggests). This uses a similar approach but explicitly incorporates velocity, solving the first-timestep problem in the Basic Verlet algorithm:

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t) \Delta t + \frac{1}{2} \vec{a}(t) \Delta t^2$$
$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\vec{a}(t) + \vec{a}(t + \Delta t)}{2} \Delta t$$

It can be shown that the error on the Velocity Verlet is of the same order as the Basic Verlet. Note that the Velocity algorithm is not necessarily more memory consuming, because it's not necessary to keep track of the velocity at every timestep during the simulation. The standard implementation scheme of this algorithm is:

1. Calculate: $\vec{v}\left(t + \frac{1}{2} \Delta t\right) = \vec{v}(t) + \frac{1}{2} \vec{a}(t) \Delta t$
2. Calculate: $\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}\left(t + \frac{1}{2} \Delta t\right) \Delta t$
3. Derive $\vec{a}(t + \Delta t)$ from the interaction potential using $\vec{x}(t + \Delta t)$
4. Calculate: $\vec{v}(t + \Delta t) = \vec{v}\left(t + \frac{1}{2} \Delta t\right) + \frac{1}{2} \vec{a}(t + \Delta t) \Delta t,$

Eliminating the half-step velocity, this algorithm may be shortened to

1. Calculate: $\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t) \Delta t + \frac{1}{2} \vec{a}(t) \Delta t^2$
2. Derive $\vec{a}(t + \Delta t)$ from the interaction potential using $\vec{x}(t + \Delta t)$
3. Calculate: $\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{1}{2} (\vec{a}(t) + \vec{a}(t + \Delta t)) \Delta t$

Note, however, that this algorithm assumes that acceleration $\vec{a}(t + \Delta t)$ only depends on position $\vec{x}(t + \Delta t)$, and does not depend on velocity $\vec{v}(t + \Delta t)$.



Molecular Dynamics in the Open



Search

GO



About OpenMD

Download

Documentation

Community

Credits

Re-use the code

Examples

News



Download and Build OpenMD



OpenMD Lab
(examples, etc.)



Ask Questions,
Share Tips, Get Help



Re-use the code



Read the Manual

What is OpenMD?

OpenMD is an open source molecular dynamics engine which is capable of efficiently simulating liquids, proteins, nanoparticles, interfaces, and other complex systems using atom types with *orientational* degrees of freedom (e.g. “sticky” atoms, point dipoles, and coarse-grained assemblies). Proteins, zeolites, lipids, transition metals (bulk, flat interfaces, and nanoparticles) have all been simulated using force fields included with the code. OpenMD works on parallel computers using the Message Passing Interface (MPI), and comes with a number of analysis and utility programs that are easy to use and modify. An OpenMD simulation is specified using a very simple meta-data language that is easy to learn.

```
◀ ▶ water_fcc.md ×

<OpenMD version=1>
  <MetaData>
    #include "water.md"

    component{
      type = "TIP4P";
      nMol = 256;
    }

    ensemble = NVE;
    forceField = "DUFF";
    cutoffMethod = "shifted_force";
    cutoffRadius = 9.0;
    dampingAlpha = 0.2;

    targetTemp = 300;
    targetPressure = 1.0;

    tauThermostat = 1e3;
    tauBarostat = 1e4;

    dt = 2.0;
    runTime = 1e4;
    useInitialTime = "false";
    useInitialExtendedSystemState = "false";

    //tempSet = "true";
    //thermalTime = 10;
    sampleTime = 100;
```

How to Connect “Physical
Experimental Observation” to
“Computational Experimental
Observation”?

Use Case: Computational Hydrology?

ADCIRC

[Community](#) [Documentation](#) [Related software](#) [News](#) [Products](#) [ASGS](#)

[Home](#)

ADCIRC

ADCIRC is a system of computer programs for solving time dependent, free surface circulation and transport problems in two and three dimensions. These programs utilize the finite element method in space allowing the use of highly flexible, unstructured grids. Typical ADCIRC applications have included:

- prediction of storm surge and flooding
- modeling tides and wind driven circulation
- larval transport studies
- near shore marine operations
- dredging feasibility and material disposal studies



[Home](#) [People](#) [Projects](#) [Models](#) [Publications](#) [Facilities](#) [ADCIRC Community](#) [Software](#) [Links of Interest](#) [Media](#) [Teaching](#) [Contact](#)

- [Overview](#)
- [Founding Developers of ADCIRC](#)
- [Advancing the State-of-the-Art in Coastal Simulations](#)

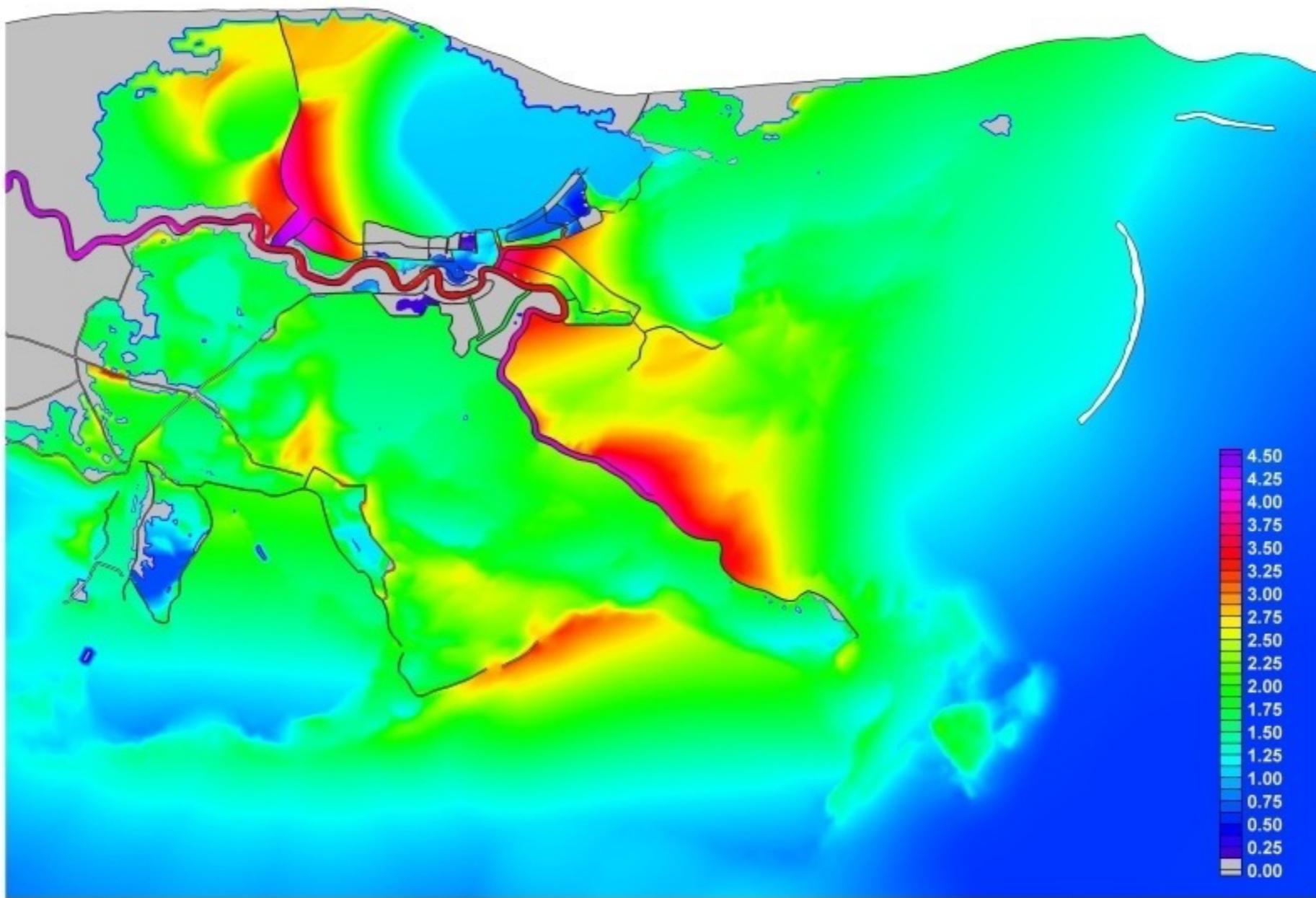
Understanding Coastal Hydrodynamic Processes and Mitigating Risk through High Fidelity Computer Simulations

Overview



The Computational Hydraulics Laboratory (CHL) at the University of Notre Dame is focused on developing cutting edge technologies to simulate wave and circulation environments in the coastal ocean and adjacent coastal floodplain. Through the development of high resolution coastal circulation and wave models, CHL has played a pivotal role in the development of hurricane and extra-tropical storm hazard modeling and has shaped today's state-of-the-art high fidelity and high performance codes and methodologies that are applied by the U.S. Army Corps of Engineers (USACE), the Federal Emergency Management Agency (FEMA), the National Oceanic and Atmospheric Administration (NOAA), the U.S. Nuclear Regulatory Commission (NRC), the U.S.

ADCIRC Storm Surge Hurricane Betsy 1965



Some (ODP) building
blocks...

SSN Ontology

Semantic Sensor Network Ontology

Author

W3C Semantic Sensor Network Incubator Group

Description

This ontology describes sensors and observations, and related concepts. It does not describe domain concepts, time, locations, etc. these are intended to be included from other ontologies via OWL imports.

This ontology is developed by the W3C Semantic Sensor Networks Incubator Group (SSN-XG). The concepts and structure of the ontology were discussed in the group's meetings and on the mailing list. For more information on the group's activities see:
<http://www.w3.org/2005/Incubator/ssn/>

Please report any errors to the Semantic Sensor Network Incubator Activity via the public W3C list public-xg-ssn@w3.org

See also

<http://www.w3.org/2005/Incubator/ssn/>

Latest Version

<http://purl.oclc.org/NET/ssnx/ssn>

Rights

Copyright 2009 - 2011 W3C.

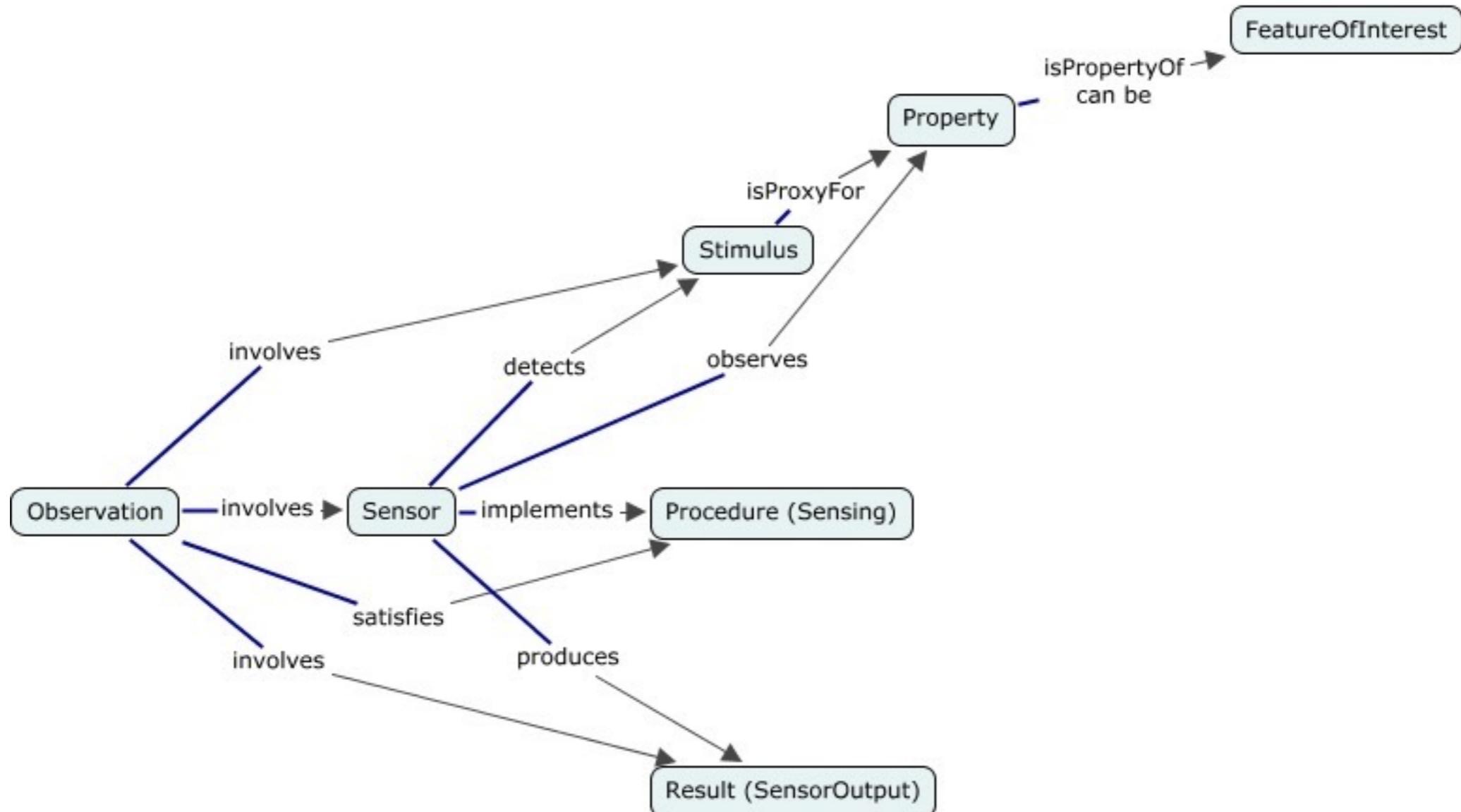
URI

The namespace for this ontology is <http://purl.oclc.org/NET/ssnx/ssn>

Prefix

When used in XML documents the suggested prefix is `ssn`

The Stimulus-Sensor-Observation Ontology Design Pattern



http://www.w3.org/2005/Incubator/ssn/wiki/SSN_Skeleton#The_Stimulus-Sensor-Observation_Ontology_Design_Pattern



PROV-O: The PROV Ontology

W3C Recommendation 30 April 2013

This version:

<http://www.w3.org/TR/2013/REC-prov-o-20130430/>

Latest published version:

<http://www.w3.org/TR/prov-o/>

Implementation report:

<http://www.w3.org/TR/2013/NOTE-prov-implementations-20130430/>

Previous version:

<http://www.w3.org/TR/2013/PR-prov-o-20130312/>

Editors:

[Timothy Lebo](#), Rensselaer Polytechnic Institute, USA

[Satya Sahoo](#), Case Western Reserve University, USA

[Deborah McGuinness](#), Rensselaer Polytechnic Institute, USA

Contributors:

(In alphabetical order)

[Khalid Belhajjame](#), University of Manchester, UK

[James Cheney](#), University of Edinburgh, UK

[David Corsar](#), University of Aberdeen, UK

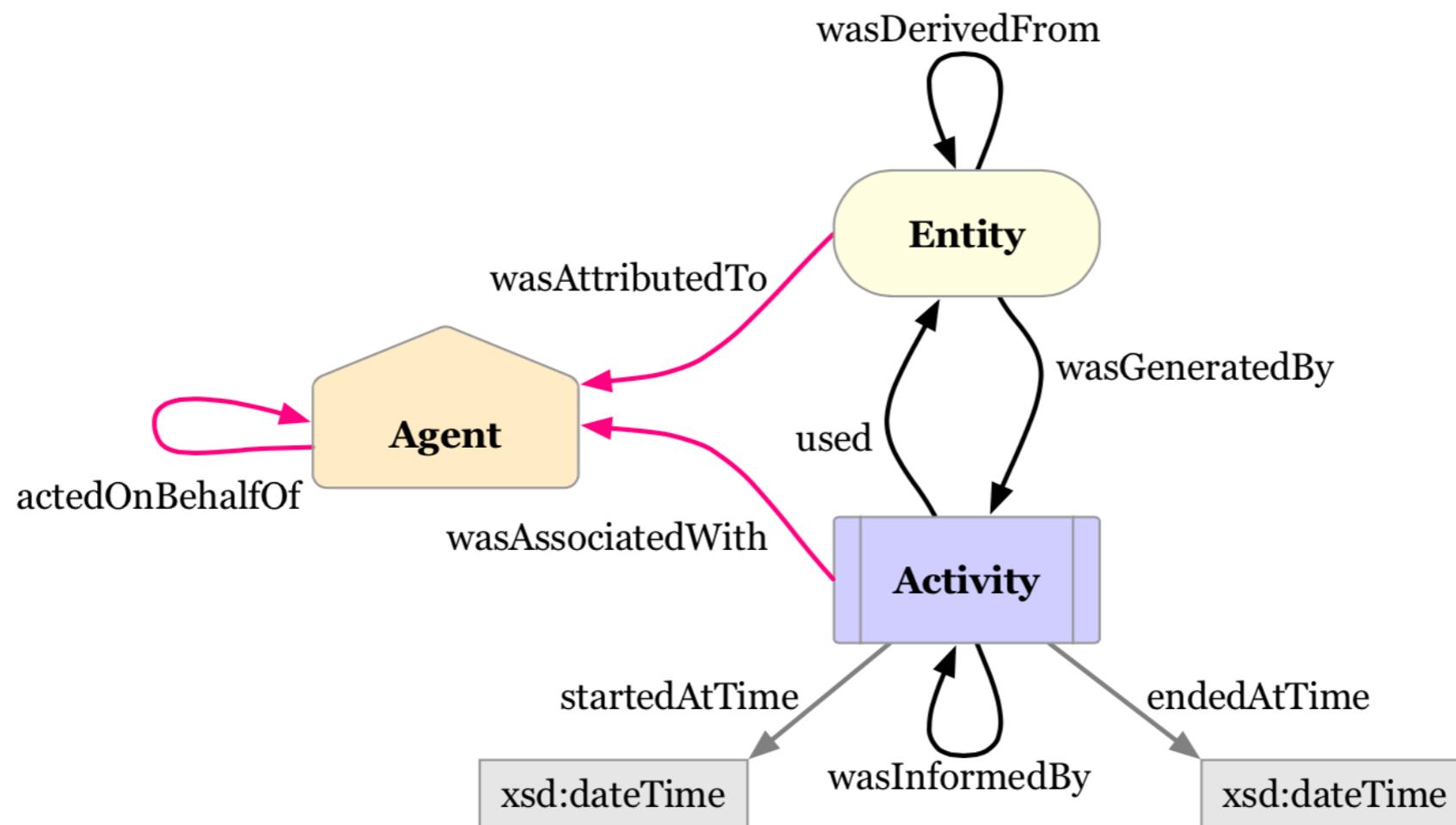
[Daniel Garijo](#), Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

[Stian Soiland-Reyes](#), University of Manchester, UK

[Stephan Zednik](#), Rensselaer Polytechnic Institute, USA

[Jun Zhao](#), University of Oxford, UK

Please refer to the [errata](#) for this document, which may include some normative corrections.



[Figure 1](#). The three Starting Point classes and the properties that relate them.

The diagrams in this document depict Entities as yellow ovals,
Activities as blue rectangles, and Agents as orange pentagons.

The responsibility properties are shown in pink.

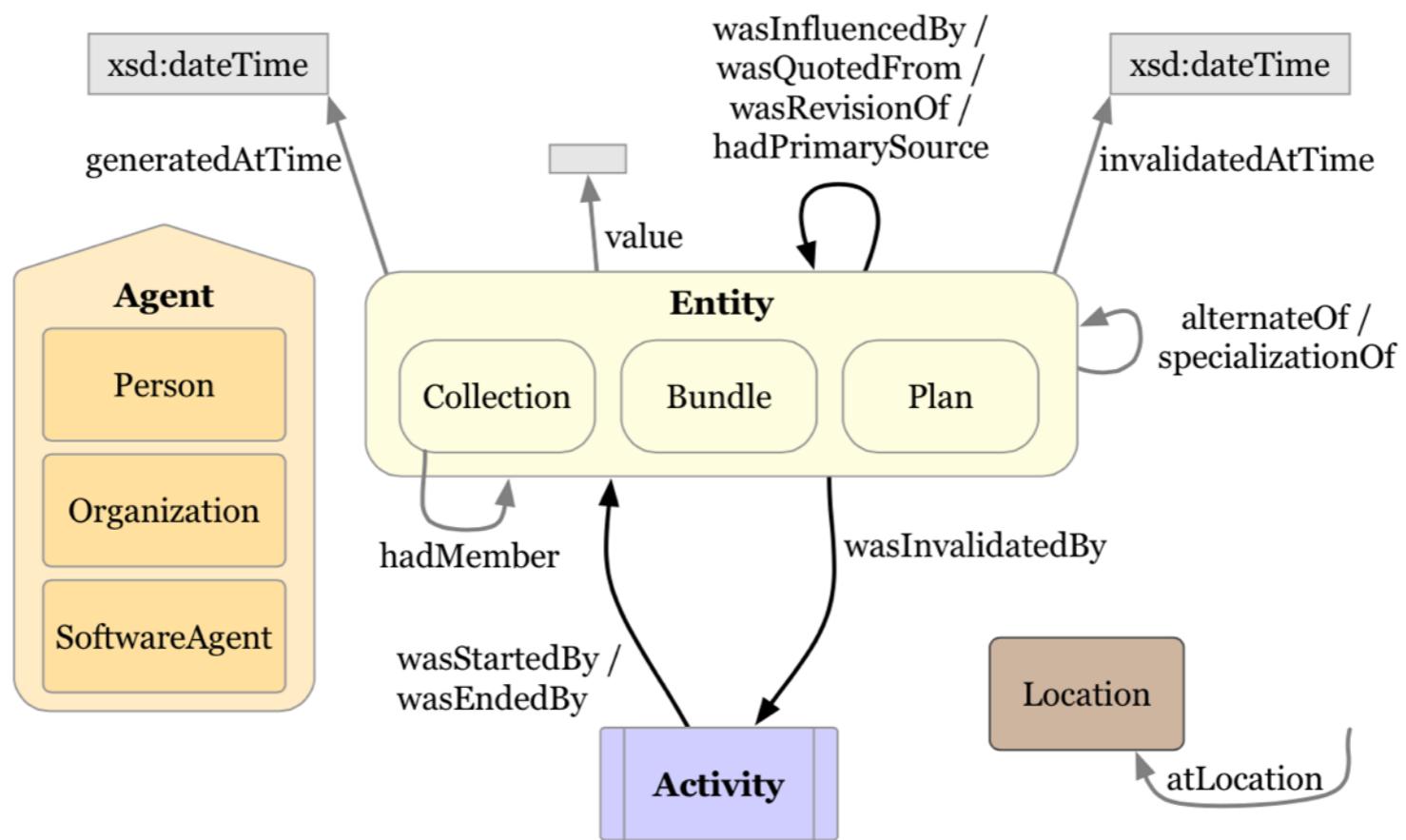


Figure 3. The expanded terms build upon those in the [Starting Points section](#).

The diagrams in this document depict Entities as yellow ovals, Activities as blue rectangles, and Agents as orange pentagons.
 The domain of [prov:atLocation](#) (`prov:Activity OR prov:Entity OR prov:Agent OR prov:InstantaneousEvent`) is not illustrated.

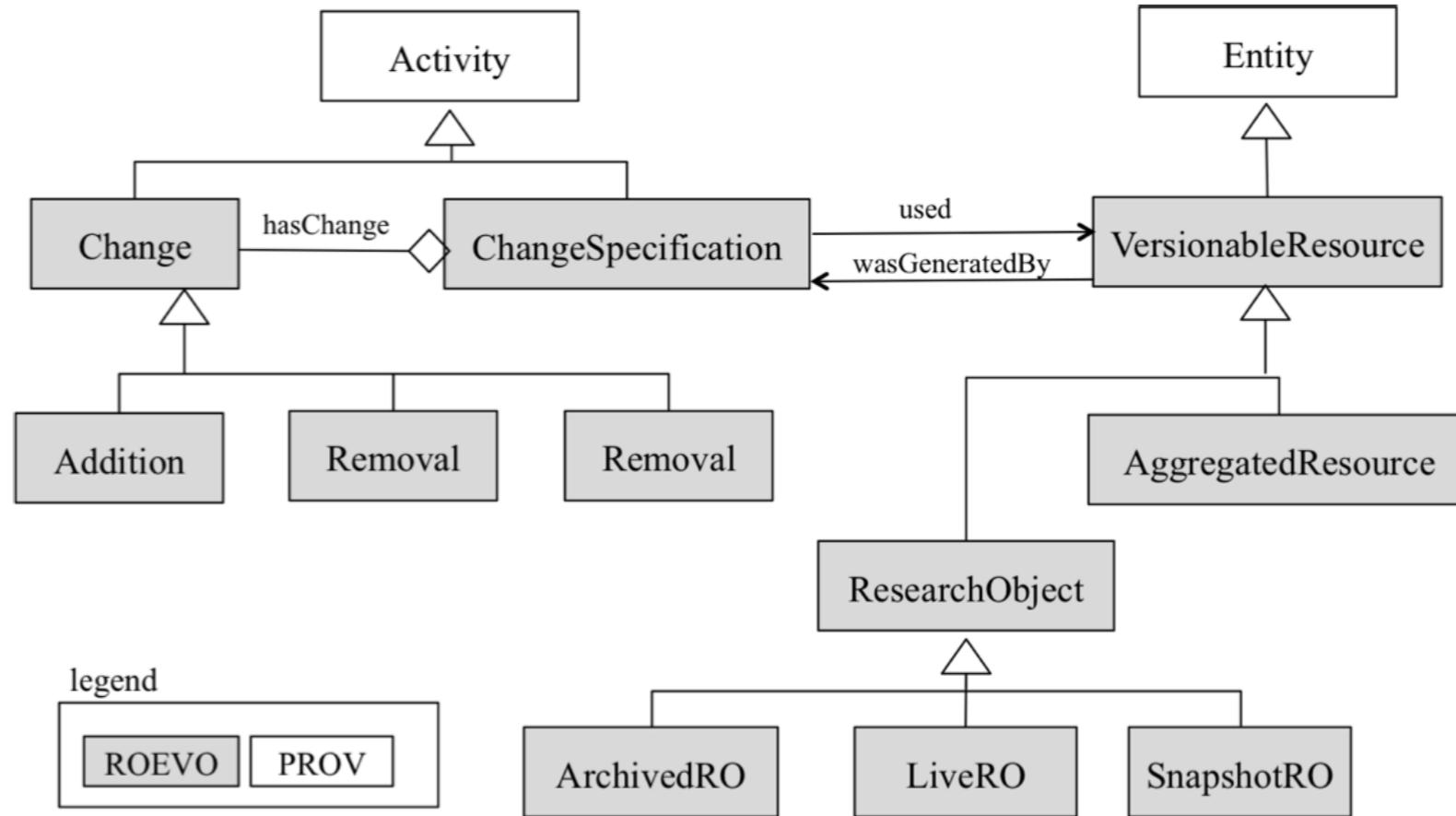


Fig. 4. The *roevo* ontology extending PROV-O terms.

The research object suite of ontologies: Sharing and exchanging research data and methods on the open web

Khalid Belhajjame, Jun Zhao, Daniel Garijo, Kristina Hettne, Raul Palma, Óscar Corcho, José-Manuel Gómez-Pérez, Sean Bechhofer, Graham Klyne, Carole Goble
arXiv preprint arXiv:1401.4307. 2014

Sensor Data Provenance: SSNO and PROV-O Together at Last

Michael Compton¹, David Corsar², and Kerry Taylor^{1,3}

¹ CSIRO Digital Productivity and Services, Canberra
`firstname.lastname@csiro.au`

² University of Aberdeen, Aberdeen, UK `dcorsar@abdn.ac.uk`

³ Australian National University

Abstract. This paper presents an alignment between the W3C Provenance Working Group’s recommended ontology (PROV-O) and the W3C Semantic Sensor Networks Incubator Group’s ontology (SSNO). The alignment views PROV-O as an upper ontology which is extended with SSNO concepts and properties. This allows representation of observation details and sensor deployments that are not possible in the SSNO alone, and gives a basis for alignment with Open Geospatial Consortium Observations & Measurements aligned ontologies. Further to the alignment, rules are presented that further constrain the interpretation of the aligned ontologies and provide a mechanism by which provenance information can be generated from SSN data thereby allowing modellers to take advantage of the new features. The benefit of the aligned ontologies is illustrated with an example of cross-domain provenance querying enabled by the alignment.

S. STAAB
R. STUDER
Editors

International
Handbooks
on Information
Systems

HANDBOOK ON ONTOLOGIES

Second Edition



Springer

An Ontology for Software

Daniel Oberle¹, Stephan Grimm², and Steffen Staab³

¹ SAP Research, CEC Karlsruhe, 76131 Karlsruhe, Germany, d.oberle@sap.com

² Research Center for Information Technology, FZI, 76131 Karlsruhe, Germany,
grimm@fzi.de

³ University of Koblenz-Landau, ISWeb, 56016 Koblenz, Germany,
staab@uni-koblenz.de

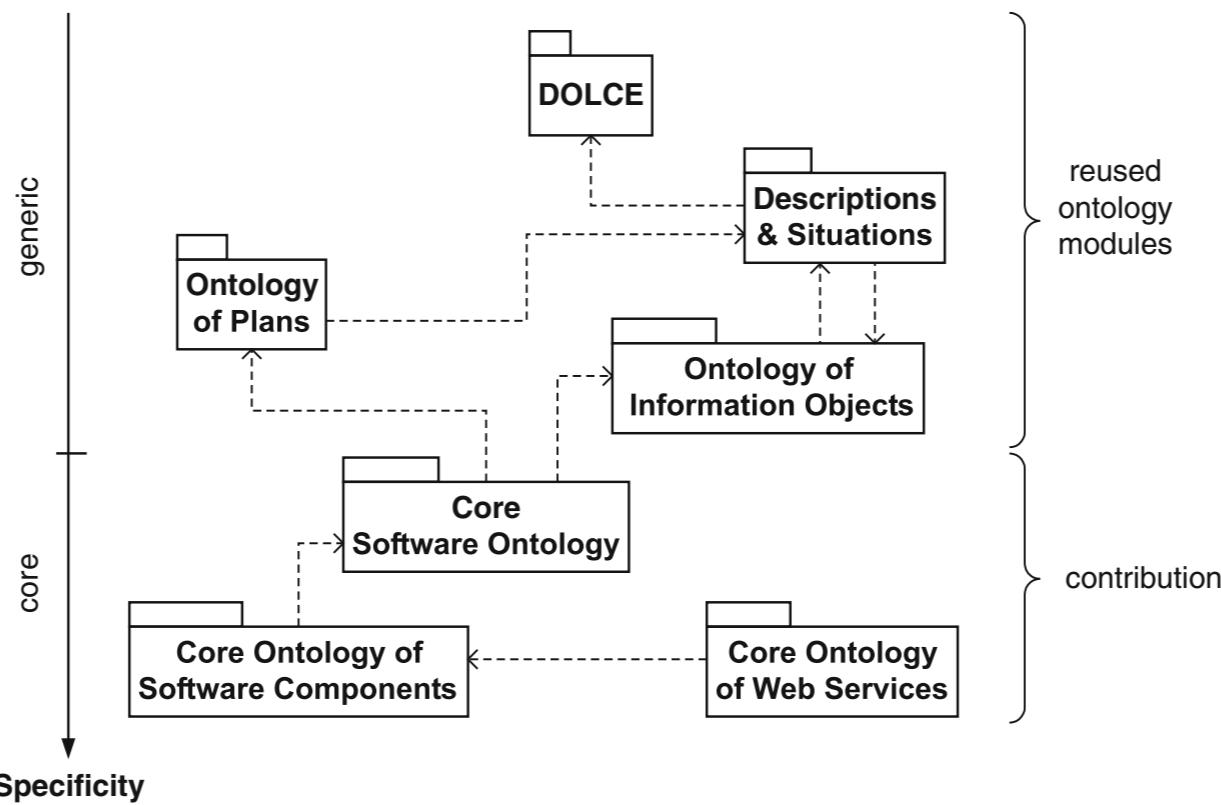


Fig. 1. Overview of the ontologies as UML package diagram. Packages represent ontologies; dotted lines represent dependencies between ontologies. An ontology O_1 depends on O_2 if it specializes concepts of O_2 , has associations with domains and ranges to O_2 or reuses its axioms

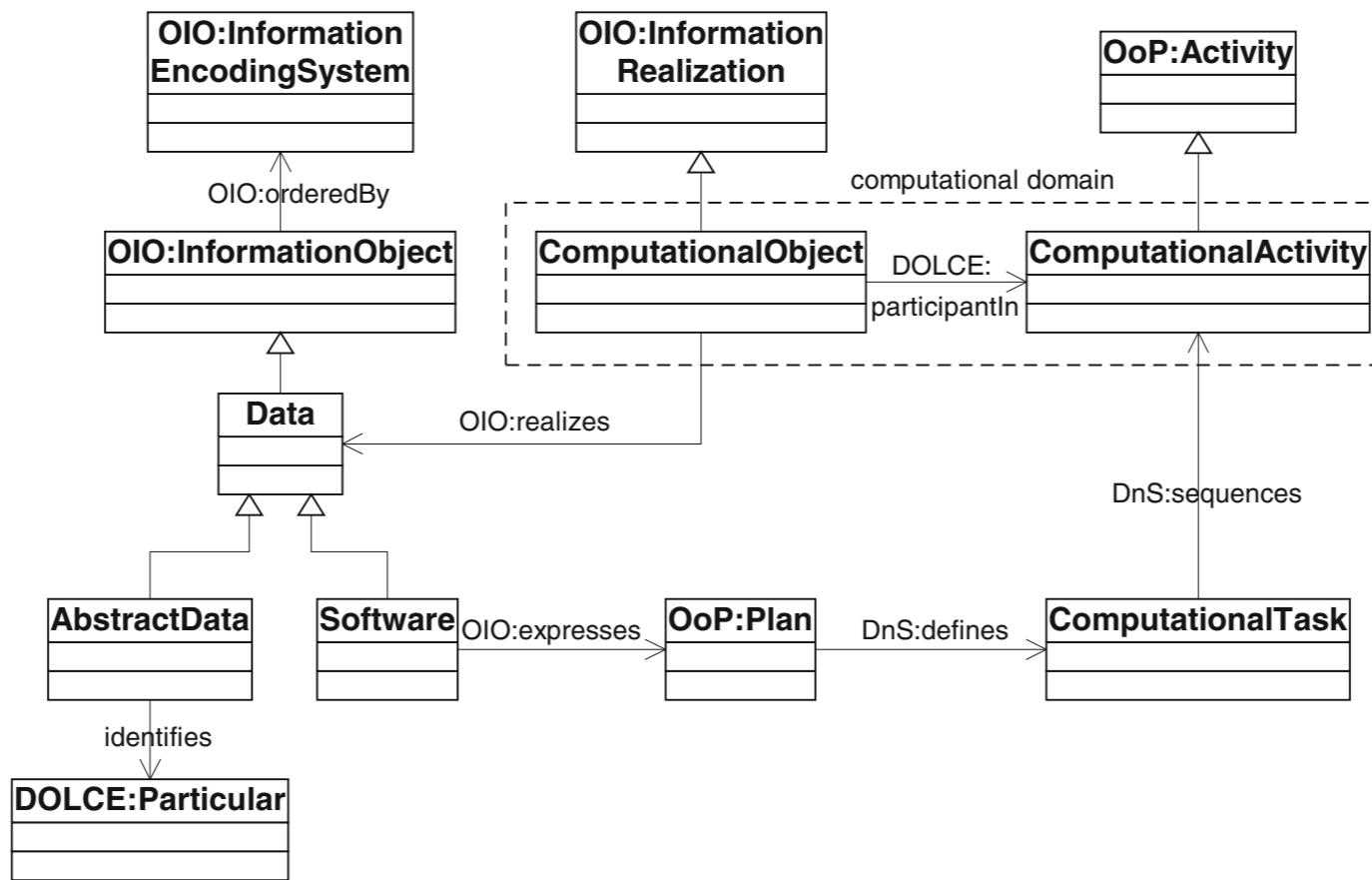
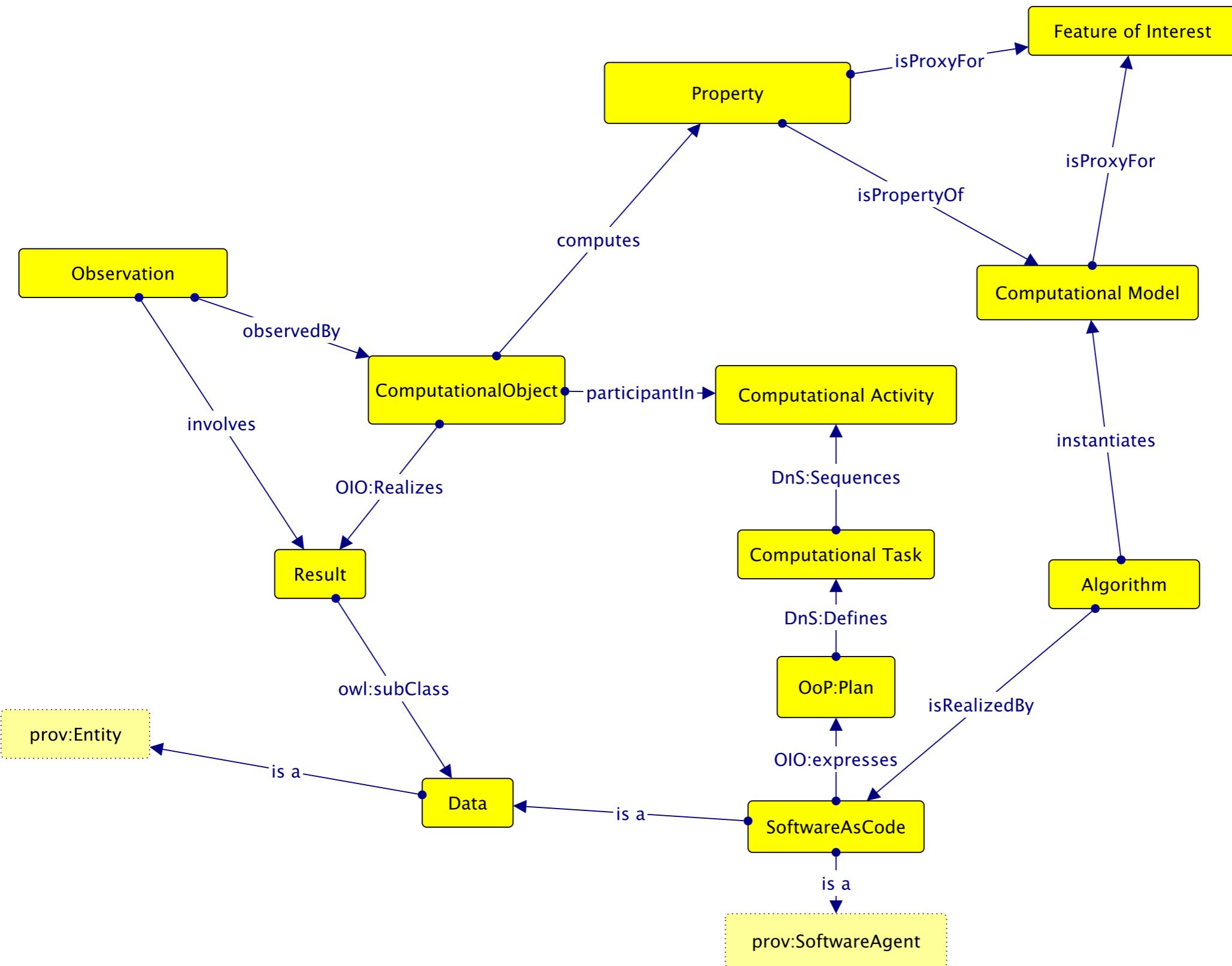


Fig. 2. The classification of software and data. Concepts and associations taken from DOLCE, Descriptions & Situations (DnS), the Ontology of Plans (OoP), the Ontology of Information Objects (OIO) are labelled with a namespace

Missing Pieces:
Conceptual and
Mathematical Model?

A straw man pattern



Thank You