Ontology for Molecular Dynamics Advanced Process Simulation

Johan Christian Edvardsen

December 1, 2017

Contents

1 Introduction

This project was done as a part of the course *Advanced Process Control* at NTNU.
The ontology simulator project that is currently being worked on aims to generate a component-based toolbox that can be used as a modelling tool that generates executable code [1]. This model code aims to be well documented, flexible, and reusable [2]. In order to assure this, the modelling tools are defined in an ontology that uses inheritance to specialize into different model networks that communicate with each other. Currently the model networks consists of the physical and signal networks, this project aims to extend the ontology to capture molecular dynamic model networks.

The small building blocks that construct network of networks are seen as complete primary models that connect with each other [2]. The goal for the ontology simulator is therefore to be able to quickly generate consistent models using a complete set of tools that guarantees a solution for any constructed model that is developed using the simulator.

This project will begin by explaining the ontology that has already been constructed and the models it capture, then molecular dynamics will be introduced and an ontology will be proposed. In the end the ontology will be introduced into the existing ontology to create a new reformed ontology.

2 Theory

2.1 Ontology

Ontology as a concept has existed in philosophy since ancient Greece and has mainly been characterized as the study of what is [3], its most general features, and how it relates to other things that exist. In other words, ontology is a systematic account of existence [4]. Although the applied concept of ontology has been used since the birth of philosophy, the word itself came into existence at a later stage.

In modernity, with the rise of AI, the semantic web, and Big Data, the ontology term has been to some degree delineated from its philosophical roots and has come to be used in these fields as a way of expressing explicit specifications of a shared conceptualization [4] in a formal way. A shared conceptualization in this sense, is an abstraction of the world that exists in some domain or field of knowledge, e.g. how the concept of an atom is viewed depends on the theory that describes it. This shared conceptualization is then made explicit by clearly defining all relevant specifications that pertains to it. These explicit definitions are afterwards formalized in such a way that they are machine understandable (readable and interpreted correctly). The use of the word ontology is therefore apt, and strays not too far from the philosophical tradition. While philosophy is concerned with that which is in human experience, information science is concerned with creating what the computer sees as what is (the computers "experience").

The motivation behind the use of ontology is the utility it brings when it is used to systematize large amounts of data based on data types and the relationships between them. Another motivation is that if an agent (in this case a computer) commits to an ontology that is coherent and consistent, then that agent is guaranteed consistency, although not necessarily completeness (as an ontology not necessarily covers everything). This also forms a basis for knowledge sharing between agents that have the same ontological commitment [4].

2.2 The Existing Ontology

In this project the ontology that has been designed is based on an already existing ontology structure and is therefore trying to extend the existing ontology. The existing ontology is used in an ontology-

based modeling tool that produces executable code [2].

2.2.1 The Model

In order to construct an ontology it must be based on a shared conceptualization. The shared conceptualization for the existing ontology is process and control modelling. The ontology is therefore intended to capture certain physical, chemical, and biological processes and the control of these processes [5]. Situating these fields into the general modelling practices found in process engineering, what emerges is a large amount of processes that can be explained in terms of control volumes and the transport between them.

One example is a chemical plant. At the largest scale the whole plant can be seen as a control volume. The transport between the plant and the environment in terms of the extensive quantities gives us a simple model that provides insight into the accumulation of mass and energy in the plant. The plant can however be divided into smaller control volumes, these could be sections of the plant or unit operations in the plant. These units can again be divided into even smaller control volumes, like trays in a distillation column. Further division can be between separate phases like the vapor and liquid phase at each tray. The picture that emerges is one where control volumes and the transfer between them can be used as a model in a large variety of scales [5].

The environment can also be divided into parts, these parts are seen as infinitely big capacities compared to the capacities (control volumes) of interest in the plant. The division separates the environment into different kinds of reservoirs, which can be seen as infinite volumes with constant properties. The environmental division into parts is therefore based on difference in property. This difference in property is important as this characterized what is fed into and extracted away from different parts of the plant.

The important difference between the plant and the environment is that the model should capture the dynamics of the plant, while the environment is modeled as constant relative to the plant. The infinite capacity of the environment is therefore not necessarily just an assumption about its volume, but an assumption about its time scale. That is, the "dynamics" of the environment is so slow that the faster dynamics of the plant "observe" the environment as constant at the given time-scale.

There are also phenomena in the plant that could be modeled as happening on a faster time scale than the dynamics of interest, e.g. a fast reaction can be modeled as something that just happens compared to the slower dynamics of mass and heat accumulation in a reactor. This introduces the need for an event-dynamic time scale, something that instantaneously happens compared to the main dynamics of interest. If however, the main dynamics of interest is the fast reactions, then the changes happening in the reactor as a capacity would be seen as very slow, and therefore the reactor would become an environment with constant properties in this case.

The model therefore aims to capture different physical, chemical, and biological processes in terms of control volumes and the transfer between them, where the application (main dynamics) defines the granularity (level of detail) of the model, which is achieved based on space- and time-scale assumptions in the design of the model [5].

2.2.2 The Mathematics of the model

When solving for a model described by control volumes and the transfer between them, what is of interest are conservation and balance equations. The accumulation and flow of mass, energy, entropy, species, and momentum. These extensive quantities are the differential states of the system, which is integrated with respect to time for a lumped capacity (ordinary differential equation), and with respect to time and space for a distributed capacity (partial differential equation) [2], in order to get the extensive quantities as states.

The differential state (the accumulation) is constructed with respect to some set of transport equations describing the transport in and out of the control volume, and in some cases with respect to some set of transposition equations which act as source/sink terms.

The transport equations depends on the nature of the transportation (convection, diffusion, conduction, and radiation) and the nature of the medium in which it is transported in (material properties, constants). The transport of the extensive properties between two capacities also depends on the difference in intensive properties between two, these are usually temperature, pressure, and chemical potential [2] (or concentration).

The transposition equations mostly cover reactions, kinetics, and phase transitions. These are dependent on the nature of the system (like reaction constants) and the intensive properties of the system (e.g. kinetics are highly dependent on temperature).

The remaining equations needed to determine the intensive properties are usually equations that depend on the state through some thermodynamic relations, which is solved by a root solver. Other equations relate to geometry, or simple relations needed to close the connection between the intensive properties and the extensive state. These kind of equations can be grouped into what is called closure equations.

Control is also implemented to be able to manipulate transport based on control volume measurement. The measurement is translated through a connection network and used to derive an error which is used to simulate the control dynamics. Then an output is derived and translated through a connections network to manipulate the transport.
The mathematical information flow can then be visualized as shown below:

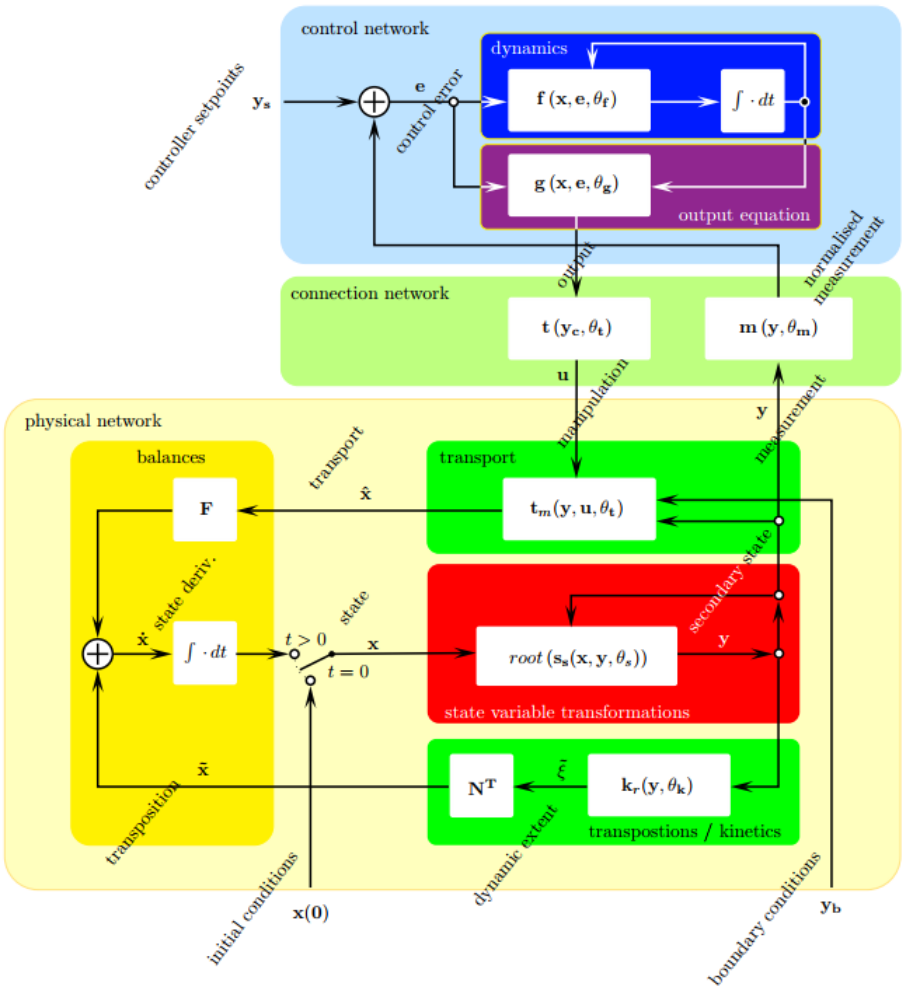

Figure 1: The mathematical model for the physical and signal domain [2]

2.2.3 The Ontology

In order to describe the constructed ontology for the control volume based models, it is necessary to look at how the models work in the abstract.

The control volume model is constructed as a directional graph that consists of capacities as nodes and the interactions between these capacities as arcs. In this graph every arc needs two nodes and every node is connected with at least one arc in order to not be trivial. The properties we want to model with this graph representation are defined as states represented by the nodes, and the properties that affect the states through interactions are represented by the arcs. That which moves around in the graph and is used to derive the states are defined as tokens. The graph is then assumed to live in time in order to capture dynamics, which is defined as the frame [1].

The ontology is constructed to capture this description, which is defined in three elements: structure, behavior, and typing. The structure defines the building blocks as abstract objects, the behavior provides mathematical descriptions to the modeled object, and the typing provides possible

structural object refinement. With this information the root ontology has been constructed as shown below:

```
[structure]  # building blocks
  graph = ['*node', '*arc']    # fixed not to be changed
  frame = ['time']

[behaviour] # link to mathematical description
  node = ['state', 'diff_state','constant','network']

[typing]  # specialisation of building block
  graph = ['physical', 'signal']
  node  = ['dynamic_0D','event', 'constant']
  arc   = ['bi-directional', 'uni-directional' ]
```

The structure in the root ontology defines a graph object that consists of a list of node objects and a list of arc objects, and the frame defines the graph as existing in time. The behavior relates the node to possible mathematical representations: state, differential state, constant, and network. The typing specialize the node into the time-scale assumptions, the arc based on directionality, and the graph into the physical or the control network.

The graph typing give rise to extensions of the ontology, one that captures the physical and one that captures control. These extensions are constructed in the same way as the root ontology with structure, behavior, and typing. Further, the extensions inherits the information defined in the root, and the extensions can communicate with each other. This communication separates the ontology from a hierarchical taxonomy (also a form of ontology), and captures communication networks that is represented as the union of two networks (e.g. the union of physical and control) [1].
The physical ontology is:

```
[structure]
  frame = ['r_x','r_y','r_z']
  token = ['mass','energy','entropy']

[behaviour]
  node = ['closure', 'transposition']
  arc = ['transport']

[typing]
  graph = ['gas', 'liquid']
  mechanism = {'mass' : ['v','d'], 'energy': ['c','w','r']}
  mass = ['*!species']
```

The structure part of the physical ontology defines the spatial directions as an extension of the frame, which is necessary to capture distributed capacities, and the ontology also introduce tokens that can consist of mass, energy, or entropy, which is used to derive the states in the physical network [1]. The behavior extends the mathematical representation of the node into the transposition and closure equations and the arc into physical transportation equations. Typing further specializes the graph into phases, the arc into different types of mass and energy transport, and the mass into a list of species.
The signal ontology is:

```
[structure]
  token = ['signal']

[behaviour]

[typing]
  signal = ['*input', '*output']
```

The signal ontology only defines in structure the token as a signal (the state is constructed based on this signal), and further types this signal as either input or output. The resulting inheritance structure of the ontology:

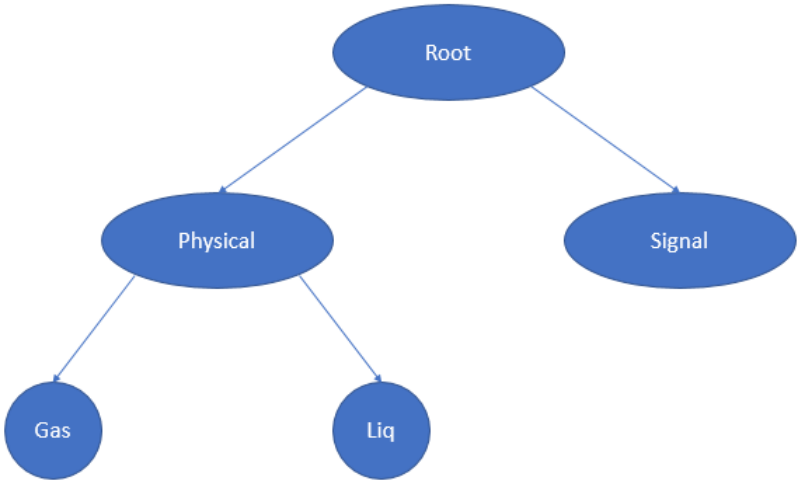

Figure 2: The inheritance structure of the existing ontology

2.3 Newtonian Physics and Molecular Dynamics

The constructed ontology tries to implement Newtonian physics with molecular dynamics into the already existing process ontology. In order to reason about how the ontology should be and what it should capture it is necessary to explore the shared conceptualization it tries to capture (Newtonian physics and molecular dynamics).

2.3.1 Newtonian Physics

With respect to Newtonian physics, it seems reasonable to try and be able to capture models that base themselves on Newton's three fundamental laws of motion:
1. An object will remain still or travel with constant velocity in a straight line if no force is acting upon the object, or the sum of forces acting upon the object equals zero,
2. The acceleration of an object is proportional with the force acting on the object and inverse proportional to the mass of the object,
3. For two objects acting on each other the force acting on the first object is of equal magnitude and with opposite direction of the force acting on the second object. The first and second law is with respect to an inertial frame of reference [6].

Keeping the first and third law in mind, we look closer at the second law with constant mass:

$$\boldsymbol{F} = \boldsymbol{M}\boldsymbol{a} = \boldsymbol{M}\frac{d\boldsymbol{v}}{dt} = \boldsymbol{M}\frac{d^2\boldsymbol{r}}{dt^2} \tag{1}$$

Where $\boldsymbol{r}$ would be the position of each object in one to three spatial dimensions(1D-3D), $\boldsymbol{v}$ are the velocities, $\boldsymbol{a}$ are the accelerations, and $\boldsymbol{F}$ are the forces. $\boldsymbol{M}$ is a diagonal (assumed constant) mass-matrix containing the total mass for each object, each mass repeated for every dimension included.

We separate the second order differential into a system of two first order differentials:

$$\frac{d\boldsymbol{v}}{dt} = \boldsymbol{M}^{-1}\boldsymbol{F} = \boldsymbol{a} \tag{2}$$

$$\frac{d\boldsymbol{r}}{dt} = \boldsymbol{v} \tag{3}$$

For this system of equations both position and velocity are defined as differential states and can be solved to find their corresponding state. At this stage the objects that can be modeled are seen as points containing mass. These points also have positions, velocities, and accelerations. The force needed to accelerate a point is proportional to its mass.

If the model is extended to include objects with volumes, it is necessary to introduce information about each object's geometry. And to further develop a model of macroscopic functional interest it is necessary to include hit-detection based on geometry, conservation of linear and angular momentum, and conservation of energy. In terms of mathematics this will introduce cross products and therefore trigonometry. Other helpful concepts such as center of mass based on density distributions in the object is also introduced in more advanced models.

In the models that will concern this project, the most important properties are conservation of total linear momentum and conservation of energy, and the view that objects are points with mass. Linear momentum is defined as:

$$\boldsymbol{p} = \boldsymbol{M}\boldsymbol{v} \tag{4}$$

The conservation of total linear momentum is then:

$$\sum_i ||\boldsymbol{p_i}|| = \sum_i m_i \, ||\boldsymbol{v}_i|| = 0 \tag{5}$$

For every object, i.
While the total and kinetic energies are:

$$E_{tot} = E_k + E_p \tag{6}$$

$$E_k = \frac{1}{2}\sum_i m_i \, ||\boldsymbol{v}_i||^2 \tag{7}$$

While potential energy can contain many things, in this project it will be a potential function that describes different kind of bonds between atoms.

2.3.2 Molecular Dynamics

Molecular dynamics is heavily based on Newton's equations of motion, however as an applied model do not concern itself with certain aspects that has been outlined above. Molecular dynamics models everything in terms of atoms as points, and simulations are done in a box with periodic boundary

conditions. Because of this the angular momentum is not conserved due to discontinuities (therefore not of interest) and as a consequence only total linear momentum is conserved and applied in calculations.

Molecular dynamics sees atoms as point-particles with masses, positions, velocities, and accelerations. The interactions between atoms are based on the negative gradient of potential functions which are introduced through the force term in Newton's second law.
The potential function is[7]:

$$E_p = V(\boldsymbol{r}^N) = \sum_{bonds,i} V_i(l(\boldsymbol{r})) + \sum_{angles,j} V_j(\theta(\boldsymbol{r})) + \sum_{torsions,k} V_k(\omega(\boldsymbol{r})) + \sum_{m=1}^{N} \sum_{n=m+1}^{N} (V_{m,n}^{LJ}(\boldsymbol{r}) + V_{m,n}^{C}(\boldsymbol{r})) \tag{8}$$

The potential function contains the bond-stretch potential, angle-bend potential, torsion potential, and the Lennard Jones and Coulomb potentials. The forces that the atoms influence each other with is then:

$$\boldsymbol{F}^N = -\nabla_{\boldsymbol{r}} V(\boldsymbol{r}^N) \tag{9}$$

Remembering Newton's third law the force between an atom-pair has this relation:

$$\boldsymbol{f}_{i,j} = -\boldsymbol{f}_{j,i} \tag{10}$$

In this projects' model, the highest complexity will be diatomic molecules with no Coulomb potential (the ontology is valid for higher complexity, but this is just to keep it simpler), resulting in potential functions on these forms[7]:

$$V(\boldsymbol{r}^N) = \sum_{bonds,i} V_i(l(\boldsymbol{r})) + \sum_{j=1}^{N} \sum_{k=j+1}^{N} V_{j,k}^{LJ}(\boldsymbol{r}) \tag{11}$$

$$V_i(l) = \frac{k_i}{2}(l_i - l_{i,0})^2 \tag{12}$$

$$V_{j,k}^{LJ}(\boldsymbol{r}) = 4\epsilon_{j,k}\left[\left(\frac{\sigma_{j,k}}{r_{j,k}}\right)^{12} - \left(\frac{\sigma_{j,k}}{r_{j,k}}\right)^{6}\right] \tag{13}$$

The force term for one dimension between two atoms that are not bonded becomes[7]:

$$f_{x,j,k} = \frac{r_{x,j} - r_{x,k}}{r_{j,k}^2}\left(24\epsilon_{j,k}\left[2\left(\frac{\sigma_{j,k}}{r_{j,k}}\right)^{12} - \left(\frac{\sigma_{j,k}}{r_{j,k}}\right)^{6}\right]\right) \tag{14}$$

For bonded atoms, the bonded force term becomes[7]:

$$f_{x,j,k} = -\frac{r_{x,j} - r_{x,k}}{r_{j,k}}(k_i(r_{j,k} - r_{j,k,0})) \tag{15}$$

The length between atoms is discerned from their positions:

$$r_{j,k} = ||\boldsymbol{r}_j - \boldsymbol{r}_k|| = \sqrt{(r_{x,j} - r_{x,k})^2 + (r_{y,j} - r_{y,k})^2 + (r_{z,j} - r_{z,k})^2} \tag{16}$$

From these equations in relation with Newton's second law it is immediately obvious that the acceleration is determined by the force, which is determined by the negative gradient of the potential function, which is again determined by the position-vector. The only thing that remains is to identify what atoms are connected with bonds, as this determines what kind of potential function each atom has.

3 Ontology for Molecular Dynamics

The constructed ontology is trying to fit molecular dynamics into the existing ontology that has been mentioned in chapter 2.2.3. Adhering to the formalism that has been outlined, that consists of: structure, behavior, and typing. The presented ontology will try to capture the kinds of Newtonian physics and molecular dynamics that has been mentioned. The ontology presented below shows all the elements in the constructed ontology, afterwards the complete ontology is shown as a combination of the existing and constructed ontology.

Observing the molecular dynamic model at its most abstract; the atoms can be seen as nodes existing in a graph that lives in time and space, the force interactions between atoms can be seen as arcs that are typed based on what interactions are molecular bonds or not. Momentum and kinetic energy is exchanged between the atoms and can be defined as tokens. The mathematical description of the atoms is that they have position and velocity as states, where one is the differential state of the other, and acceleration is the differential state of the velocity. Every pair of atoms are connected with arcs that mathematically represents the force between these atoms which is connected to their acceleration.

Information can be exchanged between a reservoir (the macro-physical system that is simulated by the molecular dynamics) and the nodes, usually in terms of velocity-correction or velocity averaging. With the information about the mathematics from section 2.3, the mathematical information flow can be seen like this (simplified model):

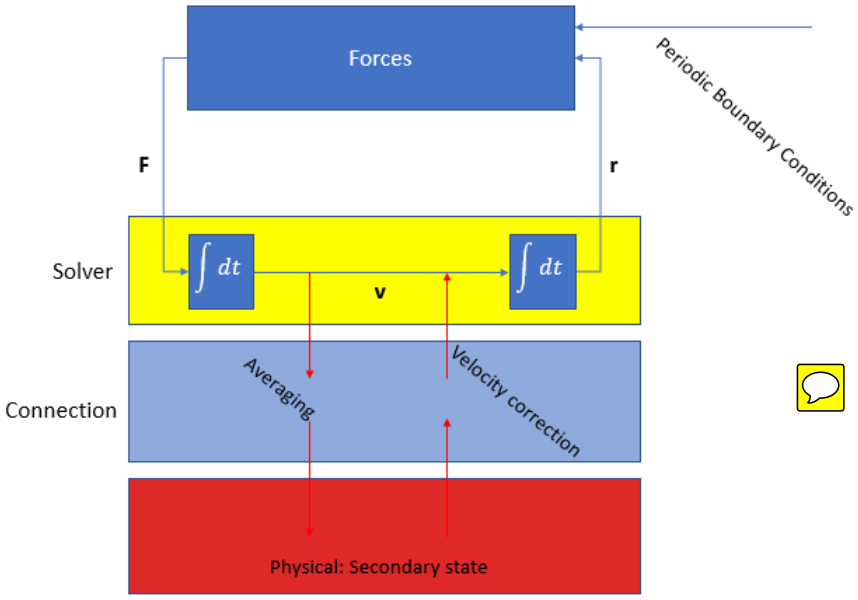

Figure 3: A simple mathematical model for molecular dynamics with connections to the physical

In the constructed pseudo-ontology shown below, the structure defines a graph object that contains a list of nodes and arcs, a frame that defines the graph as existing in time and space, and a token that defines that which flows in the graph as momentum and energy. The behavior defines the nodes mathematical state (position, velocity), differential state (velocity, acceleration), constant

(mass), and network (node-arc matrix). It also defines the arcs mathematical force which is the force calculations between nodes, and constant (constants in force). The typing defines the force further into intra- and inter molecular forces between atoms, capturing what kind of force applies between certain atoms. The nodes are of dynamic type, while arcs are bi-directional (Newtons third law). The pseudo-ontology is:

```
[structure]
  graph = ['*node', '*arc']
  frame = ['time', 'r_x', 'r_y', 'r_z']
  token = ['momentum', 'energy']

[behaviour]
  node = ['state', 'diff_state', 'constant', 'network']
  arc = ['constant', 'force']

[typing]
  force = ['intra', 'inter']
  node = ['dynamic_0D']
  arc = ['bi-directional']
```

When implementing the constructed pseudo-ontology into the existing ontology there occur changes at every level of the ontology, as well as the inclusion of new layers to keep inheritance.
The new root ontology become:

```
[structure]
  graph = ['*node', '*arc']
  frame = ['time']

[behaviour]
  node = ['state', 'diff_state','constant','network']
  arc = ['constant']

[typing]
  graph = ['physical', 'process']
  node  = ['dynamic_0D']
```

Certain changes were made due to limitations on the node and arc typing, and the inclusion of a new daughter ontology.

The physical ontology changes into the physical with typing into the macro-physical and the micro-physical ontology. The new physical ontology contains in structure the frame which defines the three spatial coordinates, token with momentum and energy. Typing types the graph into macro or micro scale, where macro is concerned with the original physical ontology and the micro concerns the molecular dynamics ontology. The arc is also here typed into 'bi-directional', which is needed as only bi-directional interactions happen at the molecular dynamics scale.
The new physical ontology is therefore:

```
[structure]
  frame = ['r_x', 'r_y', 'r_z']
  token = ['momentum', 'energy']
```

```
[behaviour]

[typing]
  graph = ['micro-physical', 'macro-physical']
  arc   = ['bi-directional']
```

Another new ontology that is currently dubbed as the process ontology follows under the root ontology and specializes into the macro-physical ontology and the signal ontology. It also types the node further into the time scale assumptions constant and event. This is necessary as the molecular dynamic system do not utilize event or constant nodes. The arc is typed into uni-directional, as this is the only arc-type signals use, and is an arc type that does not make sense in molecular dynamics because of Newtons third law.
The process ontology is therefore:

```
[structure]

[behaviour]

[typing]
  graph = ['macro-physical','signal']
  node = ['event','constant']
  arc = ['uni-directional']
```

Observe that the macro-physical ontology inherits from both the physical ontology and the process ontology.
The macro-physical ontology is:

```
[structure]
  token = ['mass','entropy']

[behaviour]
  node = ['closure', 'transposition']
  arc = ['transport']

[typing]
  graph = ['gas', 'liquid']
  transport = {'mass' : ['v','d'], 'energy': ['c','w','r']}
  mass = ['*!species']
```

This is captures the old physical ontology.

The signal ontology remains the same as in the old ontology, it has only changed place in the hierarchy.

The final micro-physical ontology inherits from the physical, and is used to capture molecular dynamics.
The micro-physical ontology becomes:

```
[structure]

[behaviour]
  arc = ['force']
```

```
[typing]
  force = ['intra-force', 'inter-force']
```

Visualizing the inheritance of the ontology, gives this tree structure:

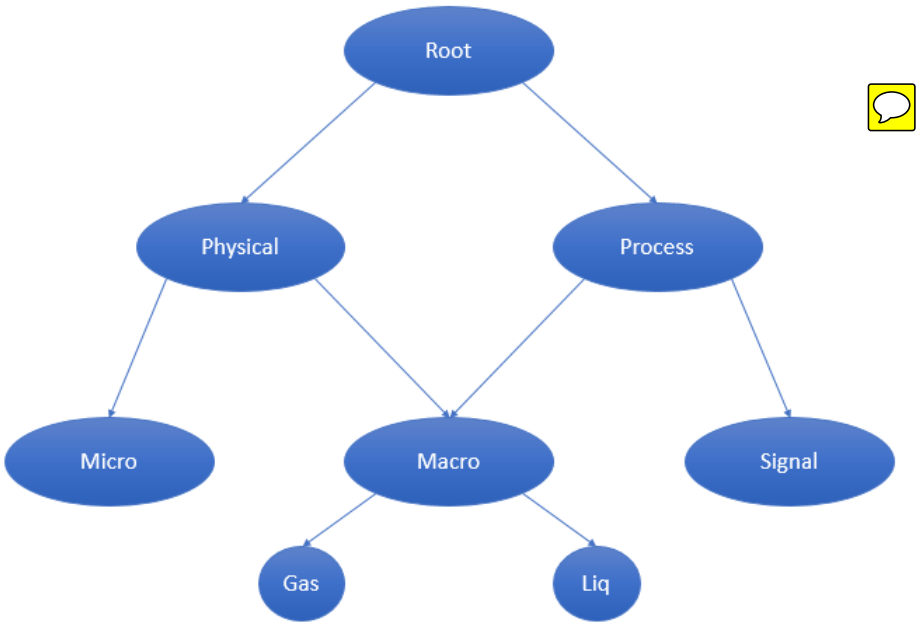

Figure 4: The inheritance structure of the new ontology

4 Discussion

Generally there are many ways of developing an ontology, the constructed molecular dynamics ontology is one of these ways. Although it is probably not the best way, it is however an ontology that can be refined and expanded upon. The placement of the constructed ontology in the new ontology is mainly based on its shared ontological definitions with the physical.

There are possibilities and problems that emerge with the new ontology. The union between the micro-physics and the signal network is beneficial, as molecular dynamics usually are controlled in some manner based on macroscopic properties, which is defined in the macro-physical network. A macro-physical network could therefore communicate with the micro-physical network through the signal network based on probability distributions derived from macroscopic properties, i.e. probability distribution of velocity in the micro-physical based on the temperature in the macro-physical (a thermostat). Extracting information the other way could also be done in terms of averaging procedures of the microscopic quantities, such as averaging of velocity for total kinetic energy and temperature. This transformation of information between networks is however not currently captured in the ontology, and is something that could be examined at a later stage. However, doing this makes it possible to give boundary conditions to the molecular dynamics, while the physical can be given average values that it lacks or need to refine.

Given what the current ontology tries to capture, it could be possible to further extend the ontology to implement traditional Newtonian mechanics, as the states would already be accounted for in the ontology. It would however require a closer look at how forces work and how geometry could be implemented. Based on how the current ontology is structured, Newtonian mechanics may end up between the physical and the micro-physical, or besides the micro-physical with a new ontology between them and the physical.

The tokens of the old ontology is what is used to define the states of a system. The states of a molecular dynamic system is however not entirely dependent on what can be classified as tokens in such a system. The states, especially the positions, are defined from frame. This implies that the current ontology does not satisfy the criterion of being formal (machine understandable). In order to implement the current ontology, the interpretation of the ontology need to be slightly configured.

The current ontology introduce momentum as a token, this is a token that can be used to identify momentum as a state instead of velocity, but position is still needed and is still dependent on frame. Since momentum is a token that at some stage probably will be implemented in the macro-physical ontology it has been placed in the physical ontology.

The placement of the force could potentially be lifted up one step on the ontology from the micro-physical to the physical, as an implementation of an momentum balance in the macro-physical could make it natural to also implement force. This is an evaluation that should be done if momentum is to be implemented into the macro-physical.

A potential problem in the modelling of the molecular dynamics is that every node would need an arc between it and every other node. Given that molecular dynamics simulations usually require a large amount of molecules, there is a potential problem of flooding the UI with a web of arcs. And if the system exchanges information with the macro-physical, then there will be another layer of arcs between every node and the macro-physical reservoir. Based on this, the visual aspect of the model should be reconsidered.

Lastly, the current ontology is at this stage only a brain-child and still needs to be implemented into the ontology simulator. There is therefore a high probability that the current ontology will require further refinement.

5 Conclusion

An ontology was proposed in order to capture molecular dynamic models, the model sees the atoms as nodes and the forces between them as arcs. The ontology was then implemented into the existing ontology and the new ontology was refined. The new ontology ended up adding three new layers, the micro-physical, the new physical, and the process layer.

Extending the ontology by implementing Newtonian mechanics into the new ontology is something that probably can be achieved as some of its workings are already implied with the implementation of molecular dynamics.

The molecular dynamic networks that can be created seem to fit quite nicely with both the physical network and the signal network in principle. This makes it possible to find physical secondary states based on the average from the molecular dynamics. The molecular dynamics can also be controlled by a thermostat that represents the physical system.

The developed ontology still need to be tested, and the interpreter in the ontology simulator probably need to be slightly tweaked in order to construct and solve for states that do not relay on tokens. Other potential problems could relate to the modelling aspect of the molecular dynamics and its visual elements.

If the proposed ontology is coherent, then the models that follow from it should also be coherent. However, equations and solvers still need to implemented before the ontology can work.

References

[1] H. A. Preisig, A. T. Elve, 2016. *Ontology construction for multi-network models*. Z. Kravanja. Proceedings of the 26th European Symposium on Computer Aided Process Engineering june 12th - 15th, 2016, 2016 Elsevier.

[2] A. T. Elve, H. A. Preisig. *Graph-based Mathematical Modeling - Concepts*. Department of Chemical Engineering, Norwegian University of Science and Technology.

[3] Hofweber, Thomas, "Logic and Ontology", *The Stanford Encyclopedia of Philosophy* (Winter 2017 Edition), Edward N. Zalta (ed.), forthcoming URL = https://plato.stanford.edu/archives/win2017/entries/logic-ontology/

[4] T. R. Gruber, 1993. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. In International Journal Human-Computer Studies 43, p.907-928. Stanford University.

[5] H. A. Preisig, 2016. *The ABC of Modelling, Lecture Notes TKP4106 4135*. version 2.0, Norwegian University of Science and Technology (NTNU).

[6] Newton's laws of motion, wikipadia. https://en.wikipedia.org/wiki/Newton%27s_laws_of_motion, updated: 27.10.2017.

[7] A. R. Leach *Molecular Modelling: Principles and Applications*, second edition, 2001. Pearson Educated Limited.