Computer Science Journal of Moldova, vol.**23**, no.2(68), 2015

The Law of Gravitation for Ontologies and Domains of Discourse*

Vadim Ermolayev

Abstract

The idea of the presented approach is to borrow a plausible analogy of a "system law"[1] from the field of Dynamics in Mechanics – the Newton's Law of Universal Gravitation. This analogy is exploited for building the law of gravitation in dynamic systems comprising a Domain of Discourse and knowledge representations (ontologies) describing this domain. As ontology elements do not possess physical mass, this component of the gravitation law is substituted by the property of *fitness* of an ontology to the requirements of the knowledge stakeholders characteristic for the described domain. It is also argued that the implementation of the developed theoretical framework is feasible as the supporting techniques, including some software tools, already exist. As the examples of the relevant component methods and tools, the paper presents concisely the OntoElect methodology, Ontology Difference Visualizer, and Structural Difference Discovery Engine. These instruments help solve some practical problems in eliciting domain requirements, developing structural contexts for

*This paper is a revised and extended presentation of the substance of the invited talk [1] given at the 2015 Workshop on Foundations of Informatics at Chisinau on the 25^{th} of August, 2015.

[1] As remarked in [2], a system law is a rule which generalizes the behavior of some observed phenomenon within a concrete system and its given spatiotemporal context. A system law tells what behavior is expected within the system. Thus a system's law can cause change or represent a barrier to change. It can be used to predict certain aspects of the system behavior, which are based on the force, or influence it exerts on the internal environment of the system. In contrast to a natural law, a system law is neither universal nor does it need to be true, correct, etc.

the requirements, generating the mappings between these structural contexts and the target ontology, computing increments and decrements of ontology fitness based on these mappings. It is concluded that the presented framework has prospects to be applied practically for visualization and analysis of ontology changes in dynamics. Use cases for ontology refinement and anomaly detection are suggested for validation.

Keywords: Ontology, Domain, Dynamics, Gravitation, Fitness

1 Introduction

The world of knowledge representations, comprising ontologies, is by its nature a reflection of the world we live in. Dynamics in physical, social, biological contexts are the subject of study by several disciplines, where useful analogies can be sought. The findings hint about a way to identify and specify useful aspects and help offer the law to describe dynamics in ontological systems.

It is known for example from Mechanics, the branch of Physics and Engineering, that Kinematics studies the motion of an object without direct reference to the causes of this motion. Motion in this context is understood as a change of position, often compared to a reference point. In difference to Kinematics, Dynamics is concerned with forces and torques and their effect on the motion of objects. For example, in Dynamics it is analyzed why an object changes its position and due to which causes or influences the acceleration has this specific value function over time.

One of the particular kinds of forces of interest regarding a physical system is gravitation. Basically, gravitation forces are known to be expressed by the Newton's Law of Universal Gravitation [3] as proportional to the product of interacting masses and inverse to the square distance between these masses. In biological and social systems similar "forces" reflect the degree of "attraction" of a particular object to a group, habitat, etc. For knowledge representations, an analogy to the notions of mass, gravitation, force could be sought in terms of the fitness of a knowledge representation module to the requirements of the stakeholders in a Domain of Discourse or its similarity to the other modules which could be found regarding the Domain of Discourse.

This paper starts with the discussion of the notion of an ontology – one of the fundamental concepts in Knowledge Representation and Management. In this context, the property of being a "shared conceptualization" is explained in terms of the fitness to the requirements of the domain knowledge stakeholders, resulting in their commitment. The paper continues with an outline of the state of the play in the field of Ontology Change, putting a particular emphasis on ontology Dynamics versus Kinematics. Then, the fundamentals of the theory of Ontology Dynamics based on the analogy to the Newton's Law of Universal Gravitation are presented. Yet further, the paper deliberates about the techniques for implementing this theoretical ontology gravitation framework. The paper concludes with the summary of the presented work and outlines the potential applications of the presented framework in Ontology Refinement and Anomaly Detection.

2 Ontologies, Domain Requirements, Fitness, and Dynamics

An ontology is often denoted as a "formal, explicit specification of a shared conceptualization" (c.f. [4]) and this paper follows this definition. In particular it is focused on describing and exploiting the properties of being "formal" and "explicit" regarding the representation of a conceptualization (specification), and – even more importantly – the property of being "shared" regarding the conceptualization itself. It is also emphasized that the completeness of an ontology has a straightforward impact on becoming a "shared conceptualization"

Being "formal" means that an ontology has to be specified using a formally defined ontology specification language such that logical inference is enabled with respect to this artifact. To enable logical inference, such a language needs to be based on logics – so an ontology is a logical theory. Ontology is also a descriptive theory as it is developed with the purpose to describe common sense, abstract high-level notions, or a Domain of Discourse.

Following [5], an ontology is a logical descriptive theory formally denoted as a tuple $O = \langle C, P, I, T, V, \leq, \perp, \in, = \rangle$, where C is the set of *concepts* (or *classes*); P is the set of *properties* (*object* and *datatype properties*); I is the set of *individuals* (or *instances*); T is the set of *datatypes*; V is the set of *values*; $\leq$ is a *reflexive*, *anti-symmetric*, and *transitive relation* on $(C \times C) \cup (P \times P) \cup (T \times T)$ called *specialization*, that helps form partial orders on C and P called *concept hierarchy* and *property hierarchy* respectively; $\perp$ is an *irreflexive* and *symmetric relation* on $(C \times C) \cup (P \times P) \cup (T \times T)$ called *exclusion*; $\in$ is a relation over $(I \times C) \cup (V \times P)$ called *instantiation*; $=$ is a relation over $I \times P \times (I \cup V)$ called *assignment*. The sets C, P, I, T, V are pairwise disjoint. It is also assumed (c.f. [6]), that an ontology O comprises its schema S and the assertional part A:

$$O = \langle S, A \rangle ; S = \langle C, P, T \rangle ; A = \langle I, V \rangle . \quad (1)$$

Ontology schema S is also referred to as a terminological component (TBox). It contains the statements describing the concepts of O, the properties of those concepts, and the axioms over the schema constituents. The set of individuals A, also referred to as assertional component (ABox), is the set of the ground statements about the individuals and their attribution to the schema – i.e. where these individuals belong.

This paper focuses on the ontologies that describe a particular well circumscribed Domain of Discourse – classified as domain ontologies. The reason for this emphasis is that any ontology development process, including its change management or refinement, takes as an input the requirements by the subject experts in the domain of interest and produces the ontology as its output – covering those requirements correctly and to the maximal possible extent. Straightforwardly, the set of methods shaping out this process needs to comprise the mechanisms for:

- Eliciting the (change[2]) requirements from the domain knowledge stakeholders as fully as possible

[2] Change requirements are elicited in the ontology Refinement phase. In the phase of Initial Development initial requirements are collected.

- Measuring how completely the requirements were captured
- Transforming the elicited requirements to the (changes in the) ontology
- Measuring how well the result fits to the intentions of the domain knowledge stakeholders

If a methodology fails to do any of the above sufficiently well, then the commitment of the knowledge stakeholders to the output ontology will be low. So, such a product cannot be regarded as a really "shared conceptualization".

Hence, a domain ontology O_D could be regarded as a harmonized formal and explicit representation of the union of the interpretations (K) by the knowledge stakeholders $s_i \in S$ of the subject domain D. So, naïvely, we may elicit all the K-s and build the ontology of those as:

$$O_D = hrm(\bigcup_S unf_{f_j}(K_{s_i})), \tag{2}$$

where hrm is a harmonization function and unf is the transformation that maps a knowledge interpretation represented in the form f_j to the knowledge representation formalism used by the knowledge engineer (unification). Even if so, harmonization and unification functions are not easy to perform. For example, a formalism f_j for K_{s_i} could be more expressive than the ontology specification language used for coding O_D; K_{s_m} and K_{s_n} could be mutually contradictory in some parts; etc. Reality introduces more complications – mainly influencing the properties of being explicit and complete:

- K-s are **subjective**. The stakeholders interpret their domain based on their individual background knowledge and experience.
- K-s are **tacit**. The views on the domain by the subject experts are often not stated explicitly. On the contrary, some parts of those K-s are assumed, taken as evident or default, subsuming that (all) the professional community regards these assumptions

in a similar way. The tacit parts are the cause for difference in interpretations, or even misinterpretations.

- K-s are **partial**. Subject experts focus on their narrow context of professional interest and expertise, and have only a shallow coverage of the broader area within the domain. The partiality and fragmentation of their K-s is the reason for (a) contradictions between different views on the overlapping contexts; and (b) gaps in the coverage of the domain.

- K-s are **not available**. The knowledge stakeholders are not readily willing to spend their time for materializing their K-s or revealing them to knowledge engineers in another form.

In Ontology Engineering and Management the degree of the conformance of an ontology to the requirements of the domain knowledge stakeholders is regarded as its *fitness*. Measuring ontology fitness is not an easy task as one has to have: the requirements; the ontology; these two compared and difference measured. Several approaches to ontology fitness measurement are known from the literature – e.g. [7, 8]. One of these approaches has been developed as a part of the OntoElect ontology engineering methodology [9]. In OntoElect, ontology fitness to domain stakeholder requirements is understood as proportional to the ratio of positive and negative votes of these stakeholders regarding the assessed ontology. These votes are collected indirectly [9], as for example in [10], by:

- Extracting a saturated set of multi-word key terms from the statistically representative document corpus

- Detecting the most influential key terms by applying weights to the most "important" documents in the corpus

- Transforming the natural language definitions of the selected key terms to formalized structural contexts in the ontology specification language; and

- Mapping the structural contexts to the ontology

Ontologies describing realistic domains could be substantially large and complex in their structures and properties. So, the development and management of these descriptive theories call for solving several interesting research problems. As profoundly surveyed in [11], ontology change – changing an ontology in response to a certain need – is one of the most important and challenging among them. Ontology Change as a field remains to be on the research and development agenda. For example, a Google Scholar search for "Ontology Dynamics" OR "Ontology Evolution" OR "Ontology Change" yields over 5 300 papers[3]. If the search is constrained by those published after 01 January 2015 it returns 224 hits.

The term of ontology change is often used broadly – to cover several interrelated facets of the problem and covering different kinds of changes to ontologies: in response to external events; caused by translations to a different language having different expressive power; caused by the evolution of stakeholder requirements; introduced by the ontology engineer according to the evolved understanding of the domain; etc. Several research sub-fields have emerged to cope with this broad variety of change aspects. The most prominent of those are:

- Ontology **evolution** (reactive response to a change in the domain or its conceptualization)
- Ontology **refinement** (goal-directed, proactive change)
- Ontology **versioning** (enable transparent access to different versions of an ontology)
- Ontology **mapping** (identify related vocabulary elements)
- Ontology **morphing** (map between vocabularies and axioms)
- Ontology **matching** (map and measure semantic distance between vocabularies and axioms)
- Ontology **alignment** (result of matching process)

[3] As of September 2, 2015

- Ontology **translation** (to a different representation language)
- Ontology **integration/ merging** (fuse knowledge from ontologies covering similar/ identical domains)
- Ontology **debugging** – **diagnosis** and **repair** (render an ontology consistent/coherent)

The plethora of these research facets, all looking at the phenomenon of change in ontologies, gave also the birth to the Ontology Dynamics community (http://ontologydynamics.org/). It may be noticed however, that the mainstream approach, also adopted by the aforementioned community, follows more Kinematics than Dynamics. Indeed, the term of "ontology change" is referred to "the problem of deciding the modifications to perform upon an ontology in response to a certain need for change as well as the implementation of these modifications and the management of their effects in depending data, services, applications, agents or other elements" (c.f.[11]). In simple words: given the need for a change, it is decided what is changed and to what extent – i.e. if following the analogy with Mechanics, how much the position, velocity, acceleration of the object changes.

It appears that the Ontology Change does not look sufficiently deeply into the causes of a change – which is in fact the task for Ontology Dynamics. In this paper some steps are made toward laying out a foundation for filling this gap based on analyzing the (changes in the) fitness of an ontology to a particular Domain of Discource.

3 Ontology Dynamics and the Law of Gravitation

Let us now think of a system, comprising a Domain of Discourse and several ontologies describing it, as of a closed "mechanical" system. For making this analogy plausible – i.e. to be able to propose usable dynamic laws – we have to find the proper analogies to the mechanical notions of: a coordinate grid and its origin; a position, a distance, a motion; a mass; and a force (gravitation).

Let us assume that a Domain of Discourse (D) is adequately modeled by the set of all relevant requirements (R), by its knowledge stakeholders, for representing knowledge in this domain. For building a grid based on these requirements it is assumed, as pictured in Fig. 1a, that:

- All the requirements are placed in the centre of the D; and
- They are not equal in their importance – i.e. have different spheres of influence around the centre of gravitation, which is quantified using normalized scores $ns \in [0, 1]$

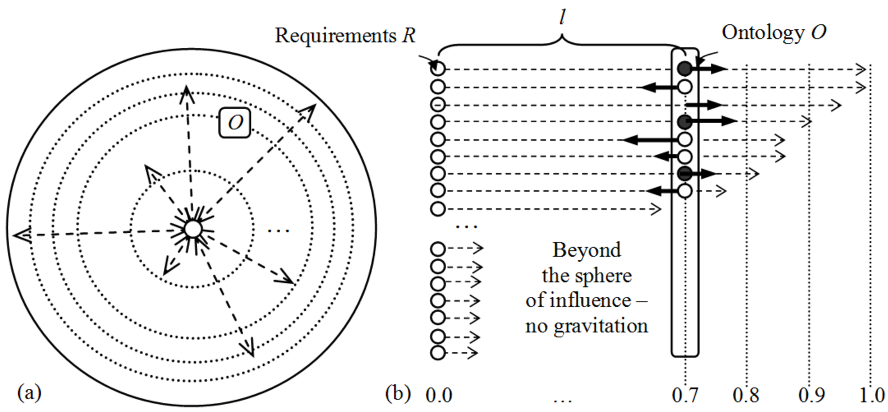


Figure 1. Domain requirements, their spheres of influence (a), and gravitation forces (b)

Let us suppose now that an ontology (O) is positioned in D at a (semantic) distance l from its centre (Fig. 1(b)). This can be any location on the circle of radius l around the centre of the grid. We are now interested in what might be the forces influencing O in this position.

Let us assume that O is checked against the requirements r from R which spheres of influence reach the position of O (i.e. $ns_r \geq l$). The following are the possible outcomes of these checks:

- A particular part of O, say a semantic context $o \in O$ (a white coloured circle in Fig. 1(b)), fulfils the requirement r. Therefore O becomes more fitting to R. In this case we will consider that the increase in fitness ($\Delta\Phi_o^+$) creates a positive gravitation force $\overrightarrow{G_o^+}$ applied to O and directed towards the centre of D, as pictured in Fig. 1(b). The absolute value of this force is computed using a direct analogy with the Newton's Law of Universal Gravitation [3]:

$$G_o^+ = \frac{1 \times \Delta\Phi_o^+}{(ns_r)^2}, \tag{3}$$

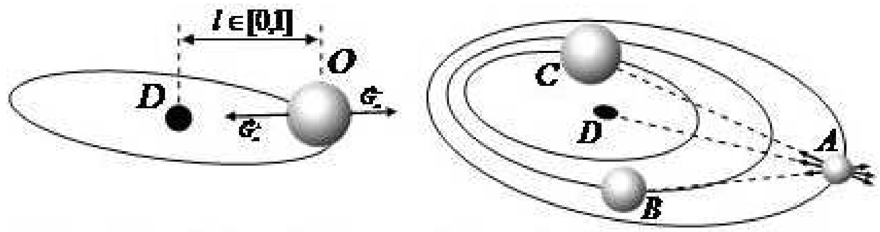
  where: "1" in the numerator is the fitness of r with respect to D – meaning that r fits D perfectly as one of its requirements; the value of $\Delta\Phi_o^+$ is within $[0, 1]$.

- There is no semantic context $o \in O$ that fulfills the requirement r (no circle on the ontology side in Fig. 1(b)) or there is an o that contradicts r (a black coloured circle in Fig. 1(b)). In both cases O becomes less fitting to R. Therefore we will consider that the decrease in fitness ($\Delta\Phi_O^-$ for a missing semantic context; $\Delta\Phi_o^-$ for a context contradictory to r) creates a negative gravitation force, $\overrightarrow{G_O^-}$ or $\overrightarrow{G_o^-}$ respectively, applied to O and directed towards the periphery of D, as pictured in Fig. 1(b). Similarly to (3), the absolute values of these forces are computed as:

$$G_O^- = \frac{1 \times \Delta\Phi_O^-}{(ns_r)^2},$$
$$G_o^- = \frac{1 \times \Delta\Phi_o^-}{(ns_r)^2}. \tag{4}$$

The overall gravitation force applied to O as an influence by D is computed as a vector sum:

$$\overrightarrow{G_O}\Big|_D = \sum_{r \in R: ns_r \geq l} \left(\overrightarrow{G_o^+} + \overrightarrow{G_O} + \overrightarrow{G_o^-} \right). \quad (5)$$



(a) One ontology in the gravitation field of *D*

(b) Several ontologies in the gravitation field of *D*, also influenced by mutual gravitation

Figure 2. Equilibrium states in the gravitation field of domain D: (a) the case of a single ontology; (b) – multiple ontologies

O is considered as *properly positioned* within D when it reaches its *equilibrium* state (Fig. 2a) with respect to the gravitation field in D, i.e. appears at a distance l from the centre of D at which $\overrightarrow{G_O}\Big|_D = \overrightarrow{0}$. This distance could be interpreted as an integral measure of the semantic difference between what does O describe and what is required to be described for D by its knowledge stakeholders. If O is not in an equilibrium state regarding D, $\overrightarrow{G_O}\Big|_D$ will cause it to move either towards the centre of D or towards its periphery. O also generates its gravitation field which affects D. However, we do not take into account the movement of D because the centre of the grid (and therefore a potential observer) is always located in the centre of D.

The gravitation field of O will come into effect in this grid if there are several ontologies positioned within D (Fig. 2b). This case is resolved similarly to the case of a single ontology described above. Ontology A reaches its *equilibrium state* within D and with respect to the ontologies B and C if $\overrightarrow{G_A}\Big|_D + \overrightarrow{G_A}\Big|_B + \overrightarrow{G_A}\Big|_C = \overrightarrow{0}$. So do the other ontologies B and C. In this equilibrium state the distances l_{AB},

l_{AC}, l_{BC} could be interpreted as the integral measures of the semantic difference in the respective pairs of ontologies, also under the influence of R in D. One topical difference for the case of multiple ontologies is that the differences and similarities in the pairs of ontologies are computed differently compared to the fitness in the pair O, D. For comparing ontologies, the use of matching techniques is the mainstream approach.

Let us now compare a mechanical system which is governed by the Newton's Gravitation Law and the proposed Domain – Ontology system using the proposed fitness-based gravitation. The comparison is summarized in Table 1.

The subsequent section focuses on the case of a single ontology in the gravitation field of D as the basic. It elaborates how the set of requirements R could be formed for D and also how ontology fitness changes could be computed.

4 Supporting Techniques

As outlined above, for making the theoretical framework based on the Law of Gravitation usable in practice several technical problems have to be solved and corresponding software tools to be developed. Let us unfold the workflow for computing gravitation forces from the outline given in Table 1.

As it may be seen in Fig. 3, the amount of work to be accomplished before the Law of Gravitation can be applied is quite high. The amount of human work may however be reduced due to:

- The re-use of the results of the previous iterations as the number of the newly coming requirements is normally much lower than of those already processed and still remaining valid for D
- The use of several instruments – methods and tools – that may help partially automate the process

The techniques and tools applicable in this context are presented in this section.

Table 1. Gravitation in a mechanical system versus domain – ontology system

	A Mechanical (e.g. Solar) System	**A Domain – Ontology System**
Coordinate grid	E.g. Helio-centric, 3 dimensional, Decartes	Domain-centric, 2 dimensional, normalized
Distance (l)	Meters, from point (0,0,0)	Normalized, semantic
Mass (m)	Kilogramms, measured using scales or other indirect methods	Fitness of O regarding R describing D
Force (G)	Newton's Law: $G = \gamma\frac{m_1 \times m_2}{l^2}$	$\left.\overrightarrow{G_O}\right\|_D = \sum\limits_{r \in R : ns_r \geq l} \left(\overrightarrow{G_o^+} + \overrightarrow{G_O^-} + \overrightarrow{G_o^-}\right)$
Model type / granularity	Continuous	Discrete
To apply	• Measure masses • Measure distance	• Extract $r \in R$ with their ns • Create knowledge tokens (kt) for r • Map kt to O • Compute $\Delta\Phi_o^+$, $\Delta\Phi_o^-$, $\Delta\Phi_O^-$

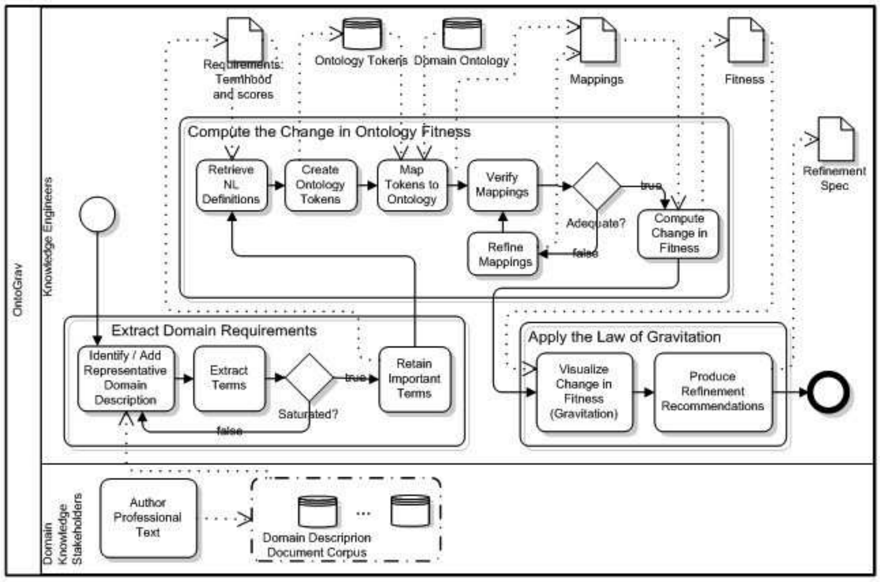


Figure 3. The workflow for applying the Law of Gravitation in the ontology refinement process

4.1 Extracting Domain Requirements

As explained in Section 2, a feasible way to make domain requirements explicit is to elicit those indirectly – by extracting multi-word key terms from a representative document corpus describing the domain. A document corpus could be considered as representative if it is sufficiently completely covers the description of the domain. One way to assess its completeness is to use the *saturation* metric proposed in OntoElect [9] as follows.

Let $Doc = Doc_1, ..., Doc_{i+1} = Doc_i \bigcup \Delta_{i+1}, ..., Doc_n$ be the sequence of the samples of the document corpus which are built incrementally – i.e. each subsequent sample Doc_{i+1} in the sequence is created by adding a number of new relevant documents (Δ_{i+1}) to the previous sample Doc_i. Let $T_i = \{(t^i_j, s^i_j, ns^i_j)\}$ be the bag of terms and their normalized scores extracted from the sample Doc_i. A normalized score

ns^i_j of a term t^i_j is computed as $ns^i_j = s^i_j / s^i_{\max}$, where $s^i_{\max}$ is the maximal score among all the terms in the bag. A bag of terms T_i is the termhood related to Doc_i if T_i contains only:

- Significant terms – i.e. those scored above the significance threshold ε_s; and
- Valid terms – i.e. those after filtering out the terms that are highly ranked, but have no substantial contribution to the semantics of the domain

One reasonable way to choose ε_s is to ensure that the terms in the termhood reflect the majority of the stakeholders' opinions. This could be done by taking in those terms from the top of the bag of terms, sorted by term score, having the sum of the scores slightly higher than the 50 per cent of the sum of all scores in the bag of terms. $Doc = Doc_1, ..., Doc_{n-1}, Doc_n$ is considered *saturated* if:

$$thd(T_{n-1}, T_n) < \varepsilon_{st}, \tag{6}$$

where: *thd* is the termhood difference function computed using the THD algorithm [9] which takes semantically equivalent and orphan terms in consideration; ε_{st} is the saturation threshold chosen empirically by a knowledge engineer for the given domain; T_{n-1}, T_n are the termhoods related to the two final document samples Doc_{n-1}, Doc_n of Doc.

It is assumed in our work that the sequence of *thd* values monotonically going down below ε_{st} indicates that Doc_n is a complete document corpus possessing sufficient representativeness. Non-monotonicity of *thd* values sequence signals that the corresponding Δ_{i+1} is either not very relevant to the domain or is a valuable addition containing the terminology not used in the previous samples (Doc_i). Anyhow, saturation indicates that the chosen document corpus is complete.

In order to apply semi-automated ontology mapping technique to compare these extracted requirements and ontology contexts, the requirements have to be represented similarly formally as the ontology. For achieving that:

- Natural language definitions for the terms in the final termhood are collected. An example is given in Figure 4. This activity is performed manually by a knowledge engineer.

- Formalized semantic contexts (knowledge tokens, *kt*) are built for the terms using the retrieved definitions. This activity could be facilitated by following the OntoElect methodology as described in [9, 10]. Knowledge tokens are in fact small ontological fragments coded in OWL DL and also visualized as UML class diagrams – as pictured in Fig. 4.

- The mappings of the constructed semantic contexts to the ontology are created. The mappings could be verified by a knowledge engineer using the visualization of the structural difference represented in an extended UML class diagram notation [14]. This activity could be done semi-automatically using the software tools for ontology alignment [13, 14].

4.2 Computing the Change in Ontology Fitness

For measuring the fitness of the entire ontology or its particular constituents with respect to the domain requirements the OntoElect methodology [9] recommends to use the metaphor of votes. Votes are computed based on:

- The scores of the respective terms t in R

- The mappings of the terms to the ontology elements

A **mapping** of the term t to ontology O is denoted as the function that establishes a relationship between t and the element of O: $\mu = (t, re, o, cf)$, where re is the relationship type – $re \in \{equivalence, membership, subsumption, meronymy, association\}$, o is the element in O, and cf is the confidence factor with a value from $[0, 1]$. Hence, $M_o = \{\mu\}$ is the set of all term mappings to the ontology element o.

A **positive vote** v_o for an ontology element $o \in O$ is denoted as a value reflecting the evidence of referring to o by the term t through the term mapping μ:

$$v_o = \sum_{\mu \in M_o} ns \times w(re) \times cf, \tag{7}$$

where: ns is the normalized score of t; cf is the confidence factor of the respective mapping μ; and $w(re)$ is the weight of the mapping based on the type of the relationship re of μ. The weights are introduced to reflect that different types of mappings could be regarded as the arguments of different strength in favour of this ontological element. Indeed, if a term is *equivalent* to the element, then it is a strong direct argument in favour of the element. However a statement about being an individual member of the element, a direct subsumption of an element, being a part of an element, or having an association to an element is considered as a weaker argument. So the weights are proposed as: *equivalence* – 1.0; *membership* – 0.7; *subsumption*, *meronymy* – 0.5; *association* – 0.3. These values may further be reconsidered if any experimental evidence is collected in this respect. Direct subsumption mappings to very abstract elements in the ontology should however be avoided. For example, all concepts, and therefore the terms categorized as concepts, subsume to the root concept of a **Thing** present in any OWL ontology. This subsumption mapping has indeed very little to do with domain semantics and therefore should not be counted as an argument for a vote. Valid direct subsumption mappings have to be sought to the most specific possible ontology elements. Indirect subsumption mappings could further be accounted for propagating votes up the concept hierarchy as described below. Propagated votes may be used to further clarify the distribution of the fitness upwards the subsumption hierarchy of the ontology.

So far only direct positive votes with respect to ontology elements have been discussed. So, the overall ontology fitness computed based on these votes reflects only the arguments focused on an element and without any influence on the surrounding of this element. This however might not be fully correct with respect to the fitness of the surround-

A **Clock** is an **Instrument** to generate the instances of a **TemporalMeasure** of a **TimeInstant** when this **TimeInstant** instance is also the instance of a **Present**. A **Clock** is always associated with a particular single **TimeLine** (there could be **TimeLines** with no **Clock** but also **TimeLines** having several different **Clocks** associated with them). Different **Clocks**, associated with the same **TimeLine** or different **TimeLines** may 'run" differently, e.g. quicker or slower and also with offsets compared to each other. Some **Clocks** may be related to each other for enabling a proper comparison of the values they return. This is done by specifying a **ClockRelation**. A specific and most widely used kind of a **ClockRelation** is **AffineClockRelation** which allows aligning different time velocities (using the scaleFactor property) and also time offsets, like delays (using the shift property). A **Clock**, as a measurement instrument, may return a single value (a **TimeStamp** corresponding to a single **TimeUnit**) or several values (the parts of a **TimeStamp** corresponding to different **TimeUnits**). A **PhysicalClock** and a **LogicalClock** are the two disjoint specializations of a **Clock**.

(a) A natural language definition of the concept of a Clock

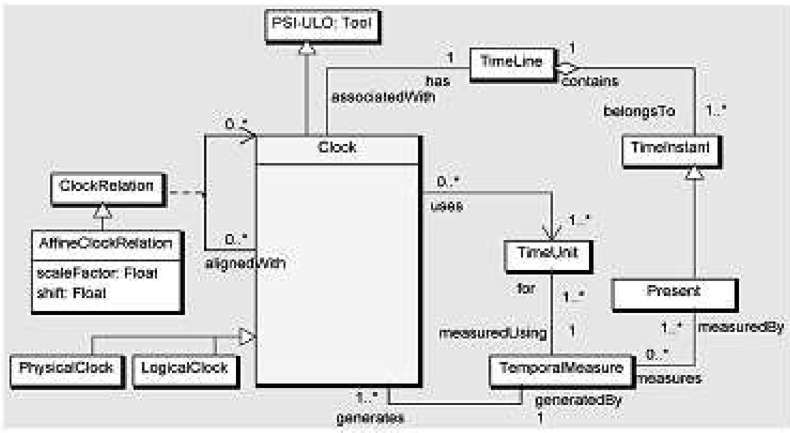


Figure 4. Term processing pipline by example. The term and semantic context of a Clock

ing elements. Indeed, let us for example assume that the concept of a **Clock** in a Time ontology gets a vote. Then it may be expected that the concept of an **Instrument**, subsuming **Clock** (See also Fig. 4), also qualifies for the part of the value of this vote. A straightforward reason is that, due to the subsumption relationship, the more specific concept inherits the properties of the more abstract concept in the subsumption hierarchy. So the vote has to be propagated up the hierarchy with attenuation – factored empirically or possibly aligned with the proportion of the inherited properties in each individual case.

A **propagated vote** v_o^p for an ontology element $o \in O$ is the value reflecting the contribution of o to the semantics of the ontology element o^{sub} subsumed by o:

$$v_o^p = att \times v_{o^{sub}}, \tag{8}$$

where att is the attenuation coefficient.

Positive and propagated votes provided by the term t are further used for computing the **fitness increments** $\Delta\Phi_o^+$ of the elements in O.

$$\Delta\Phi_o^+ = \sum_{\mu \in M_o} v_o + \sum_{O_o^{sub}} v_o^p, \tag{9}$$

where O_o^{sub} is the subset of the elements in O which are subsumed by o.

A **negative vote** provided by a term t ($v_t^- = -ns$) is:

- Either a vote based on the term $t \in T^{miss}$ pointing out that t is not described by O. In this case a fitness decrement for the whole ontology O could be computed as:

$$\Delta\Phi_O^- = v_t^- \mid_{t \in T^{miss}} ; \tag{10}$$

- Or a vote pointing out that the term t is in a contradiction with a particular ontology element o. In this case a fitness decrement for the ontology element $o \in O$ could be computed as:

$$\Delta\Phi_o^- = v_t^-. \tag{11}$$

The overall change in ontology fitness caused by the influence of the term t (requirement $r \in R$), being the sum of all positive, propagated, and negative votes could hence be computed as follows:

$$\Delta\Phi_O\,|_t = \sum_O \left(\Delta\Phi_o^+ + \Delta\Phi_o^-\right) + \Delta\Phi_O^-. \tag{12}$$

Consequently, the change in overall ontology fitness caused by R is:

$$\Delta\Phi_O\,|_R = \sum_R \left(\Delta\Phi_O\,|_t\right), \tag{13}$$

As already mentioned in Section 2.2, all these changes are taking effect if the sphere of influence ns of the requirement $r = (t, ns) \in R$ is more or equal to the distance l between O and D.

4.3 Computing Mappings between Ontologies

The creation of the mappings of the semantic contexts of the terms from the termhood (knowledge tokens, *kt*) could be done in a partially automated way using an appropriate ontology matching technique. One possible technique is meaning negotiation using argumentation based on the exchange of presuppositions [12]. This approach has been implemented in several software tools supporting different steps in the mapping generation process:

- Generation of the mappings between the TBoxes of two different ontologies in the ontology alignment format or as ABox transformation rules could be facilitated using the Structural Difference Discovery Engine (SDDE) [13]

SDDE uses an approach for ontology alignment based on the implementation of meaning negotiation [12] between intelligent software agents. Their negotiation strategy implies aligning ontologies by parts (conceptual subgraphs or contexts) that are relevant to a particular negotiation encounter. Negotiation is conducted in an iterative manner and is aimed at the reduction of a semantic distance between the contexts. Agents use propositional substitutions, expressed in a Type theory, which may reduce the distance, and support them with argumentation. The process is stopped when the distance reaches some commonly accepted threshold or the parties exhaust their propositions and arguments. The software produces a set of mappings between the ontology fragments either in the Ontology Alignment Format [15] or as transformation rules [16]. The mappings are produced as XML serializations of $\mu = (t, re, o, cf)$ – as explained in Section 4.2. These mappings, after been verified, may be refined using the Transformation Rule Editor of the OIM Tool [16] – as pictured in Fig. 5.

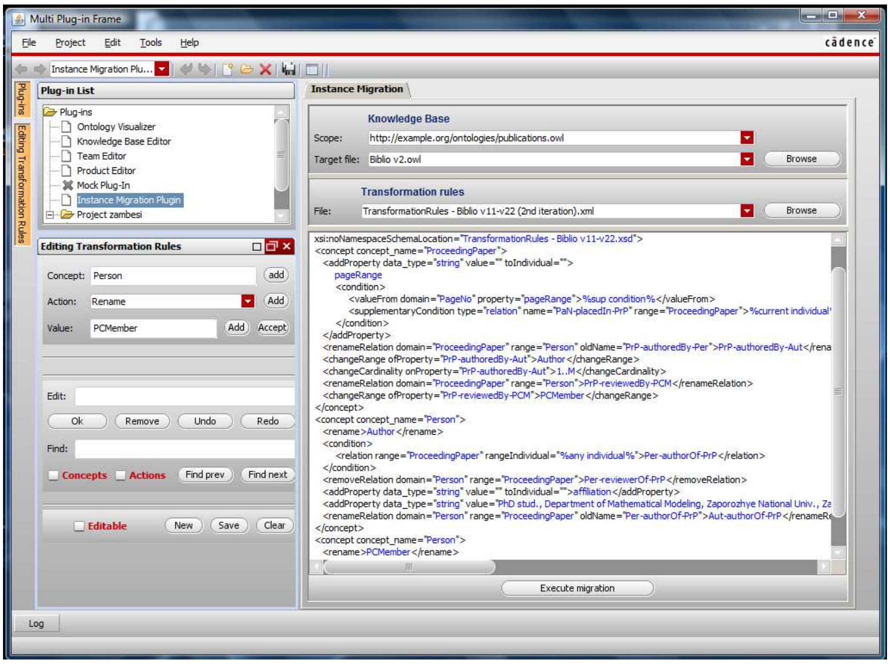


Figure 5. OIM tool Dashboard and Transformation Rule Editor, adopted from [16]

- Verifying the structural changes between the TBoxes of two different OWL ontologies by visualizing the difference using an extension to the UML class diagram language could be performed by the Ontology Difference Visualizer (ODV) tool [14]. ODV desktop is pictured in Fig. 6.

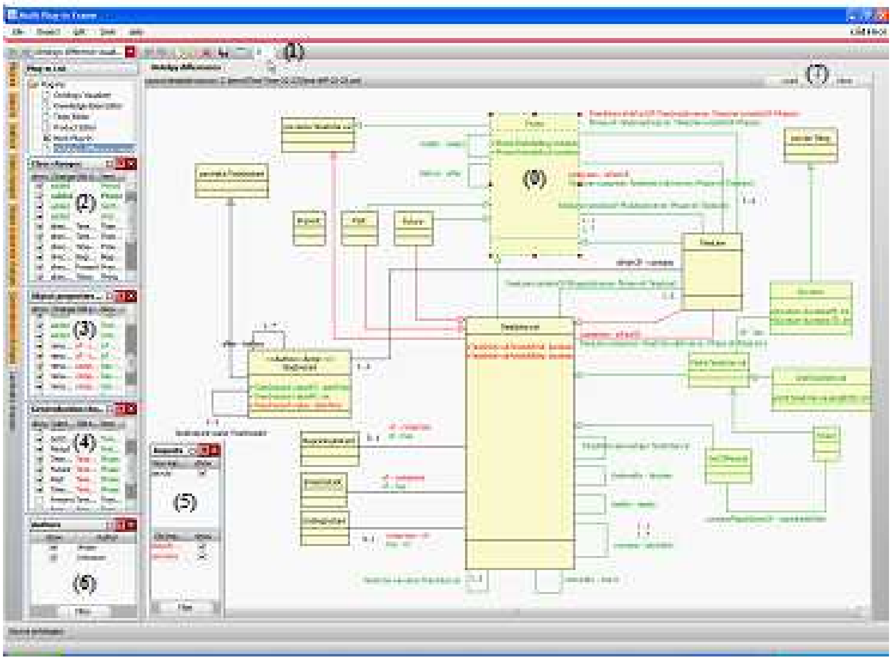


Figure 6. The ODV desktop. Visualized is the structural difference between the PSI Time Core ontologies v.2.2 and 2.3. Adopted from [14]

The composition of a semantic context of a concept (0), as implemented in the ODV, could be formed by specifying the radius of the neighborhood of this concept (1). Further it could be fine-tuned by manual inclusion or exclusion of the concepts (2), object properties (3), subsumption relationships (4). The analyzed ontological context may be placed on the wafer of the source (old) ontology by toggling the "show old" mode in the Tools menu. The context may be also altered by considering or filtering out the concepts belonging to the imported ontology modules (5). Finally the "owner" filter may be employed for concentrating on the changes that have been introduced by a particular ontology engineer in the team (6). The ODV implementation allows also editing and saving the layout of the visualized structural difference in the project file (7). Such a layout saves all the context settings and therefore allows personalized representations for different users. The release of the ODV proof of concept software prototype [14] has been implemented in Java as a plug-in to the Cadence ProjectNavigator software prototype. ODV uses OWL API 2 and therefore is capable of processing OWL ontologies coded in OWL DL.

5 Conclusive Remarks

This paper presented the approach to deal with dynamics in knowledge representations, in the form of ontologies, regarding the domains these ontologies are intended to describe. In order to place the reported research in the context of the scientific discipline, the basics of Ontology Engineering, Management, and Change have been concisely presented in Section 2.

The high-level idea followed in the presented work is to understand the dynamics of ontologies in a way similar to the other scientific disciplines – primarily answering the questions about the causes of a change and therefore offering the laws to compute forces and their effect on the motion of ontologies within the domain. Hence, the central part of the presented research deals with an attempt to exploit the analogy with the Newton's law of Universal Gravitation. This law has however to be applied to the objects that do not possess physical mass. Therefore, the proper analogues for a mass, a coordinate grid and its origin; a position, a distance, a motion; and a force (gravitation) have been elaborated – resulting in a theoretical Ontology Gravitation framework presented in Section 3. This framework is based on the notion and measurement of ontology fitness to the knowledge stakeholder requirements to the description of a particular Domain of Discourse.

It has also been described in Section 4 of the paper that the implementation of the presented theoretical framework is feasible as the supporting techniques, including some software tools already exist. The presentation focused on outlining the opportunities provided by the OntoElect methodology, Ontology Difference Visualizer, and Structural Difference Discovery Engine to help solve the practical problems in:

- Eliciting domain requirements without any direct involvement of the knowledge stakeholders
- Developing structural contexts for multiple word key phrases that indicate the requirements
- Generating the mappings between these structural contexts and the target ontology
- Computing increments and decrements of ontology fitness based on these mappings

The framework presented in the paper has prospects to be applied practically for visualization and analysis of ontology changes in dynamics. The following use cases could be of particular scientific, industrial, and societal value.

Ontology refinement is the implementation of the required changes in an ontology for making it fit the changed stakeholder requirements to the maximal possible extent – Fig. 7. In the terms of the Ontology Gravitation framework described above, stakeholder requirements are captured by R for D (Fig. 1), each having also its *ns*. So the changes in these requirements result in the changes to the gravitation field generated by D (Fig. 7a and 7b). These in turn will cause that the ontology O changes its position to reach a new equilibrium state in the changed gravitation field of D (Fig 7c and 7d). This new position of O may appear to be closer to the centre of D's gravitation – which indicates that the changes in the stakeholder requirements were favourable for the current implementation of O. It may also appear, as in Fig. 7d, that O will move further out from the gravitation centre of D – indicating that the changes in requirements hint about the necessity to refine O. A visualization tool showing the changes in ontology equilibrium state positions in response to the changes in the gravitation field of D may become a powerful instrument for a knowledge engineer to assess and justify the refinement of the particular fragments of the ontology. Such a justification will be based on the acquired knowledge, in a condensed and visualized form, about the causes triggering the needed change.

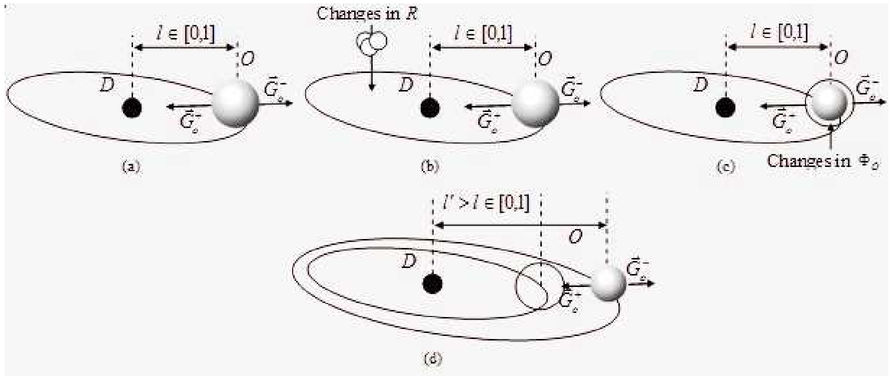


Figure 7. A way to visualize the changes in ontology fitness in ontology refinement

Anomaly detection in data analytics is about revealing the parts of data that change beyond normal values – hinting about a potential or developing problem in the system that is the source of these data. For example, if a system is a civil community and its environment (D), then it may be producing many diverse streams of observation data coming from various sorts of sensors – like outdoor temperature measurements, water levels, industrial emissions, share prices, cell phone activity, etc. Imagine that each sort of censor measurement is described by its individual ontology which is updated using knowledge extraction from the respective incoming data stream. From the other hand, community requirements R reflect the desire of the stakeholders to live in a comfortable (normal) environment: clean air and water; stable share prices, no traffic hold-ups, etc. If so, it is reasonable to expect that an equilibrium state, involving the abovementioned sensor data ontologies and D, will show how close (normal) or far (abnormal) each sort of sensor measurement is from the normal condition. This visual result may be made available in time sufficient for emergency response to the detected anomaly.

6 Acknowledgements

The theory of gravitation between ontologies and Domains of Discource, based on fitness, has been developed in the SemData project funded by the Marie Cure International Research Staff Exchange Scheme (IRSES) of the 7th Framework Programme of the European Union, grant No PIRSES-GA-2013-612551. OntoElect methodology has been developed as a part of the PhD project of Olga Tatarintseva. The tools for ontology mapping and alignment have been developed with Anton Copylov and Maxim Davidovsky.

References

[1] V. Ermolayev. *The Law of Gravitation in Ontology Dynamics.* In: S. Cojocaru and C. Gaindric (Eds.): Proc. W-shop on Foundations in Informatics (FOI 2015), pp. 246–267, Acad. of Sciences of Moldova, 2015.

[2] W.-E. Matzke. *Engineering Design Performance Management – from Alchemy to Science through ISTa (Invited Talk).* ISTA 2005, 23-25 May 2005, Palmerston North, New Zealand, pp. 154–179 (2005).

[3] I. Newton. *The Principia, Mathematical Principles of Natural Philosophy, a new Translation.* By I Bernard Cohen and Anne Whitman, preceded by "A Guide to Newton's Principia" by I Bernard Cohen, University of California Press, 1999, ISBN 978-0-520-08816-0, ISBN 978-0-520-08817-7.

[4] Rudi Studer, V. Richard Benjamins, Dieter Fensel. *Knowledge Engineering: Principles and Methods.* In: Data & Knowledge Engineering, 25(1-2):161–197, 1998.

[5] J. Euzenat, P. Shvaiko. *Ontology Matching*, Berlin Heidelberg (DE), Springer-Verlag, 2007.

[6] D. Nardi, R. J. Brachman. *An Introduction to Description Logics.* In The Description Logic Handbook, F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, Eds. Cambridge University Press New York, NY, USA, 2007.

[7] A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann. *Modelling ontology evaluation and validation.* In: Sure, Y., Domingue, J. (eds.) ESWC 2006, LNCS 4011, pp. 140–154 (2006).

[8] Fabian Neuhaus, Amanda Vizedom, Ken Baclawski, Mike Bennett, Mike Dean, Michael Denny, Michael Grüninger, Ali Hashemi, Terry Longstreth, Leo Obrst, Steve Ray, Ram Sriram, Todd Schneider, Marcela Vegetti, Matthew West, Peter Yim. *Towards Ontology Evaluation Across the Life Cycle.* In: Applied Ontology. 8 (2013), pp. 179–194, DOI 10.3233/AO-130125.

[9] O. Tatarintseva, V. Ermolayev, B. Keller, W.-E. Matzke. *Quantifying Ontology Fitness in OntoElect Using Saturation- and Vote-Based Metrics.* In: Ermolayev et al. (eds.) ICT in Education, Research, and Industrial Applications. Revised Selected Papers of ICTERI 2013, CCIS **412**, pp. 136–162, Springer Verlag, Berlin – Heidelberg (2013).

[10] V. Ermolayev, S. Batsakis, N. Keberle, O. Tatarintseva, G. Antoniou. *Ontologies of Time: Review and Trends.* In: Int. J. of Computer Science & Applications, 11(3), pp. 57–115, 2014.

[11] Giorgos Flouris, Dimitris Manakanatas, Haridimos Kondylakis, Dimitris Plexousakis, Grigoris Antoniou. *Ontology Change: Classification and Survey.* In: The Knowledge Engineering Review, 23(2), pp. 117-152, 2008. doi:10.1017/S0269888908001367.

[12] V. Ermolayev, N. Keberle, W.-E. Matzke, V. Vladimirov. *A Strategy for Automated Meaning Negotiation in Distributed Information Retrieval.* In: Y. Gil et al. (Eds.): ISWC 2005 Proc. 4th Int.

Semantic Web Conference (ISWC'05), 6-10 November, Galway, Ireland, LNCS 3729, pp. 201–215 (2005).

[13] M. Davidovsky, V. Ermolayev, V. Tolok. *Agent-Based Implementation for the Discovery of Structural Difference in OWL-DL Ontologies.* In: H.C. Mayr et al. (Eds.): UNISCON 2012, LNBIP 137, pp. 87–95 (2013).

[14] V. Ermolayev, A. Copylov, N. Keberle, E. Jentzsch, W.-E. Matzke. *Using Contexts in Ontology Structural Change Analysis.* In: Ermolayev, V., Gomez-Perez, J.-M., Haase, P., Warren, P, (eds.) CIAO 2010, CEUR-WS, vol. 626 (2010).

[15] J. David, J. Euzenat, F. Scharffe, C. Trojahn dos Santos. *The Alignment API 4.0,* Semantic Web, 2(1), 3-10 (2011).

[16] M. Davidovsky, V. Ermolayev, V. Tolok. *Instance Migration Between Ontologies having Structural Differences.* Int J on Artificial Intelligence Tools. 20(6) 1127–1156 (2011) DOI: 10.1142/S0218213011000553

Vadim Ermolayev

Received September 7, 2015

Department of IT, Zaporizhzhya National University,
66 Zhukovskogo st., 69063, Zaporizhzhya, Ukraine
E–mail: vadim@ermolayev.com