DERI – Digital Enterprise Research Institute

Ontology Maintenance: Quantifying the Conceptual Dynamics in Domain Ontologies

Martin Hepp

mhepp@computer.org

http://www.heppnetz.de

Semantics in Business Information Systems (SEBIS) Group
DERI, University of Innsbruck

DERI TECHNICAL REPORT **SEBIS-TR2007-01**

26 June 2007

Preferred citation: Hepp, Martin: *Ontology Maintenance: Quantifying the Conceptual Dynamics in Domain Ontologies*. DERI Technical Report SEBIS-TR2007-01, June 26, 2007, available at http://www.heppnetz.de.

DERI – Digital
Enterprise Research
Institute

DERI Galway
National University of Ireland,
Galway
Ireland
www.deri.ie

DERI Innsbruck
University of Innsbruck
Technikerstrasse 21a
Innsbruck
Austria
www.deri.at

DERI Korea
267 Deokil Bldg (Saltlux),
Daechi-dong, Gangnam-gu,
Seoul 135-848
Korea
http://www.deri-korea.org

DERI Stanford
Stanford University
Serra Mall
Stanford
USA
www.deri.us

Ontology Maintenance: Quantifying the Conceptual Dynamics in Domain Ontologies

Martin Hepp

Digital Enterprise Research Institute (DERI), University of Innsbruck
mhepp@computer.org

Abstract. Most practically relevant domains include some degree of conceptual dynamics, i.e. new conceptual elements arise, and some old ones become irrelevant. In the products and services domain, for instance, corporations are continuously inventing new types of goods; and in physics, scientists may discover new types of particles or relations between them. This type of change imposes a substantial challenge on building up-to-date domain ontologies. While most research on ontology evolution has focused on technical solutions for managing such change that has already materialized in the ontology *formalization*, the order of magnitude of dynamics in real-world domains as the input variable to ontology engineering has not yet been studied. In this paper, we (1) provide a generic simulation model, based on a colored Petri-net, for measuring the conceptual dynamics and the resulting domain coverage of ontologies and (2) quantify the dynamics in three selected areas, i.e. for computer components, pharmaceuticals, and for methods, recipes, and procedures in the inorganic chemical technology sector. We can show that all three areas undergo a substantial conceptual dynamics and that this may lead to weak domain coverage in respective ontologies. Based on these findings, we (3) discuss approaches of how the engineering lag of building domain ontologies can be reduced.

1. Introduction

Materializing the promises of semantics-aware systems will require high-quality domain ontologies, i.e. such that cover respective topic areas in sufficient detail so that tasks like e.g. search, content extraction, content reuse and integration, or services discovery can be supported in real-world scenarios. For many promising domains of interest, we are unfortunately still lacking up-to-date ontologies with a sufficient granularity and expressivity. Of those domain ontologies that are published on the Web, only some are actively maintained and thus reflect the current domain vocabulary. Many others are rather outdated prototypes of one-time snapshots of a domain.

Now, we can observe that most real-world domains include some degree of conceptual dynamics, i.e. that new elements arise as some old ones become irrelevant. In the products and services domain, for instance, manufacturers are continuously inventing new types of goods, and technical progress requires adding new attributes; in physics, scientists can discover new types of particles or relations among them; and in the geopolitical domain, new states form and political borders change. In some cases, one may argue whether such change is really a change in the ontology (e.g. the domain theory) or rather one of the broader knowledge base (i.e. including data). We will discuss this question in more detail in section 2, but assume that at least a part of those changes requires actually a change in the ontology. In the area of products and services, it is for instance pretty obvious that technological advancement will require new concepts or new attributes for describing novel product models or product instances.

Fensel has stressed that ontologies are the glue between real-world semantics and formal semantics [1], i.e. that they are not just formal domain theories but models of the world that must also reflect reality as *perceived* by human actors. On a philosophical level, we may argue over whether all abstract concepts exist independently of time and human discovery, e.g. whether the categories for all potential products and services that may ever be invented already exist. Practically, however, we can assume that ontology engineers and domain experts can only add such new elements to the ontology that are *known and actively used* in the respective domain. In effect, the dynamics of conceptual elements in reality or in the perception of reality is relevant when building a domain ontology for a particular area of interest.

In database research, the problems caused by domain evolution have been stressed e.g. by Ventrone and Heiler [2]. Similarly in ontology research, the challenges of change and dynamics have already been discussed by several researchers, e.g. by Noy and Klein [3], Heflin and Hendler [4], Fensel [1], Klein and Fensel [5], and Klein at al. [6]. In the field of methodologies for ontology engineering, e.g. the DILIGENT approach has put iterative maintenance and user/domain feedback to the center of building ontologies in order to deal better with change [7].

Changes in ontologies have been traced back by Noy and Klein [3] and Klein and Fensel [5] to three causes, i.e. (1) changes in the domain, (2) changes in the (shared) conceptualization, and (3) changes in the explicit specification. Most research has focused on how the change in and among evolving formal specifications can be managed, e.g. how we can maintain interoperability in a network of changing ontology specifications (in the sense of formalizations) so that instance data and ontology imports remain consistent or at least so that conflicts are minimized.

Unfortunately, the conceptual dynamics of domains as the origin of change has not yet received a lot of interest from researchers. This is in sharp contrast to the fact that such dynamics may be a significantly limiting factor when building and maintaining detailed domain ontologies. We assume that the lack of interest is likely because conceptual dynamics is less obvious when dealing with upper-level concepts such as "physical matter," "agent," or "intangible." Since, historically, finding ontological truth at a high-level of abstraction has been an important guideline of building ontologies in Computer Science, we may falsely assume that creating lasting ontologies was a mere matter of proper conceptual modeling. That is, once we have discovered a proper model of a domain of discourse, the conceptualization and specification would be stable for ages. While we fully agree that cleanliness in conceptual modeling is important for creating stable ontologies, there is evidence that this alone will not be sufficient for dealing with the dynamics faced in domain- and task-specific ontologies.

Pinto and Martins [8] were one of the few who identified dynamism as a relevant dimension of ontology engineering projects. Also, some work has been done with regard to measuring the amount of change in domain *specifications*: Klein et al. [6]

briefly reported the amount of change in the UNSPSC categorization schema, and in [9], we presented a comprehensive analysis of the amount of update operations in the four products and services classifications UNSPSC, eCl@ss, eOTD, and the RosettaNet Technical dictionary. In eCl@ss, for instance, there have been about 280 new and more than 1200 modified classes *per 30 days* (!) in versions 5.x [9]. However, it must be stressed that these two studies did not attempt to measure the *domain* dynamics but the dynamics in domain *specifications*. This is insofar relevant as the dynamics in specifications is likely much lower than that in the actual domain. After all, specifications incorporate only that subset of the overall change which has successfully passed all bottlenecks and obstacles of standardization etc. We know from [9] that even large industry classifications with more than 25,000 concepts still lack a lot of relevant concepts, and this despite the fact that the degree of formality of these specifications is very limited.

In this paper, we will use a simulation model based on a colored Petri-net [cf. 10] and data from three real-world domains for measuring the amount of conceptual dynamics in various domains and determining the amount of domain elements that are missing in a domain ontology.

1.1. Our Contribution

In this paper, we (1) provide a generic simulation model, based on a colored Petri-net, for measuring the conceptual dynamics and the resulting domain coverage of ontologies. We then quantify the dynamics in three selected areas, i.e. (2) for computer components, (3) pharmaceuticals, and (4) for methods, recipes, and procedures in the inorganic chemical technology sector. We can show that all three areas have substantial domain dynamics and that ontologies updated in typical intervals will suffer from weak domain coverage. Consequently, we (5) discuss approaches of how building ontologies in dynamic domains can be improved.

One core advantage of our Petri-net-based approach is that the structure and constraints of the ontology engineering process and the resulting delay in updating the ontology can be modeled very flexibly. It is thus possible to refine our model so that it will reflect in more detail the steps and temporal behavior of various ontology engineering methodologies. Also, a multiplicity of constraining factors can be included, e.g. such delays caused by fixed update intervals, by required lead times, or those due to a limited amount of processing “bandwidth” (e.g., “not more than 10 change requests can be processed per day”).

1.2. Structure of the Paper

In section 2, we analyze the interplay between domain dynamics, the ontology engineering and maintenance lag, and possible domain coverage of an ontology. In section 3, we present our research methodology, namely a simulation model based on a colored Petri-net. In section 4, we describe the data sources that we used for our experiments and what kind of pre-processing we carried out. In section 5, we present the results from the simulation runs and show how the amount of concepts missing in a respective domain ontology would develop over time. In section 6, we discuss and evaluate our findings and derive implications for building domain ontologies. Section 7 highlights our main results and concludes the paper.

2. Conceptual Dynamics and Ontology Maintenance Lag

In this section, we analyze how the unavoidable delay in producing a shared formalization constrains the inclusion of novel conceptual elements from the domain, and thus limits the amount of up-to-dateness of a domain ontology.

2.1. Conceptual Elements in Domain Ontologies

Ontologies are commonly specified as a set of conceptual entities of the respective domain of discourse. Those entities are usually (1) described using human readable text and (2) their interpretation is constrained by formal axioms [cf. 11, 12]. The typical elements of ontologies are classes, attributes (slots), relations, functions, and axioms. Also, instances (individuals) may belong to the ontology as long as they are "ontological" instances, i.e. such that are not mere data but a necessary part of the shared conceptualization. For example, an ontology that needs to define the class of inhabitants of Innsbruck must include the city of Innsbruck as an instance of "city" for being able to define the respective constraint (slot restriction) on the attribute "isCitizenOf". However, in a more abstract tourism and travel ontology, only "city" as a class would be part of the ontology, but the city of London would be instance *data*. It highly depends on the scope and purpose of the ontology whether a particular individual is an "ontological instance" or data of the knowledge base, and there is often also room for argument. However, there is little doubt that some instances may belong into the ontology. This is in particular true for *domain* ontologies and is insofar relevant as much of the domain dynamics takes place in the area of "ontological instances". For the remainder of the paper, we refer to all potential elements of an ontology as "conceptual elements" and mean with this all concepts/classes, attributes, relations, *ontological* instances/individuals, functions, and axioms.

2.2. A Model of Dynamics, Formalization Delay, and Domain Coverage of Ontologies

Whenever we design an ontology, we face a fundamental problem: It takes time for the involved stakeholders (1) to agree upon the relevant conceptual elements and their definition and (2) to produce a shareable ontology formalization. At the same time, new conceptual elements become relevant in a domain of discourse, which was not yet included in the initial domain capture. From an engineering perspective, it would be better if we were able to „freeze" the discourse and dynamics while we are working on the consensual model of the domain, but of course we cannot. This holds both for the initial formalization of an ontology and for consequent updates. The very same problems are known from standardization [13]. They have also recently been sketched for ontologies [14].

We can find such dynamics in almost any domain: in a sports and leisure ontology, new types of sports activities are becoming popular (e.g. "rafting", "sandboarding", "street skating", "kite snowboarding", etc.). In the legal domain, new categories of punishable acts may be defined. In biology and medicine, new classes of species may develop due to evolution or may be discovered and named. Now, it is a triviality that a domain's conceptual dynamics increases with the specificity of modeling, (a class hierarchy's granularity, for example). As long as (1) the ontology is fairly abstract, (2) the engineering and maintenance delays are small, and (3) the conceptual dynamics is limited, updating the ontology may be a lesser problem. Also, since ontologies should follow the principle of minimal ontological commitment [15] and thus be, in general, more abstract, this type of a knowledge acquisition and maintenance bottleneck is less problematic than with detailed knowledge bases. Eventually, the conceptual dynamics will be very limited when building top-level ontologies, since the microscopic everyday advancements that mankind makes usually leave the big categories of "tangible vs. intangible" or "role vs. actor" untouched.

However, it is also pretty obvious that materializing most of the promises about ontologies and the Semantic Web will require *detailed* domain ontologies in addition to top-level abstractions. For instance, if we want to use ontologies for the automated mediation between message flows from a set of incompatible legacy systems in the billing processes of telecommunications companies, then we need such ontologies that reflect all the conceptual elements in that domain: contract types and billing plans, locations, types of telecommunication equipment, etc. Likely the most striking example is the products and services domain, where new types of goods and new attributes of existing goods categories are invented or introduced on a daily basis.

One could argue that it was just a matter of fact that some novel concepts were missing in a domain ontology. Unfortunately, often the novel concepts in a domain are those for which semantic technology would be most interesting. For instance, when searching the Web for price comparison, it are the novel goods for which the price differences are likely most substantial. For long established categories, competition on the market and arbitrage will have balanced out prices, and traditional technology like e.g. numbering schemas of other types of standardization has been established and may be used to facilitate non-semantic search.

Ontologies could help tremendously when broad consensus about terminology has not yet been established or when non-semantic approaches are too slow. After all, the increase in agility in data and process integration is often considered one of the major advantages of ontologies.

The basic structure of the problem of domain dynamics on one hand and the lag in ontology creation and maintenance on the other hand is illustrated in Figure 1. The upper line depicts the amount of conceptual elements in reality, caused by the continuous "birth" of new ones. (For the moment, we abstract from the removal of outdated elements, since keeping them in the ontology does often not harm.) The lower line reflects the amount of conceptual elements that are included in the most recent release of the ontology formalization. We can easily see the fundamental problem: Once the initial domain capture for the ontology is completed (t_0), it takes some time to formalize and release the ontology. Thus, the first version of the ontology will not be available until t_1. This ontology contains all elements (classes, instances, attributes, relations, and axioms) that reflect the initial domain capture in t_0. In the meantime, however, additional conceptual elements have become relevant in the real world, as depicted by the upper line. All such new elements are not included in the ontology and can thus neither be used for annotating data nor for expressing queries.

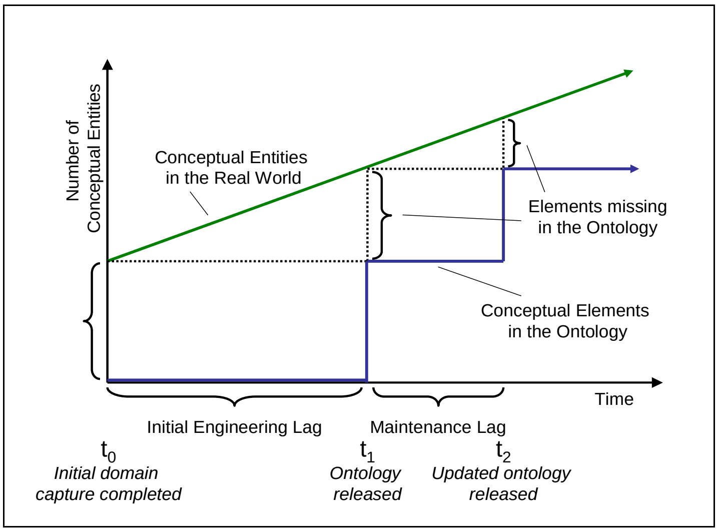

Fig. 1. Conceptual dynamics and coverage of domain ontologies

If the ontology is actively maintained, we may carry out an updated domain capture at t_1, but producing the updated ontology and documentation again takes time, making the new version available at t_2. In the meantime, however, additional conceptual elements have again become relevant in the real world, which will be missing in this ontology update. Thus, we are trapped in a vicious circle: Each time we release a new version of the ontology formalization, it may be the perfect shared conceptualization for the respective domain – but only with regard to the state in which the domain was when we completed our domain capture.

3. Methodology

In this section, we describe our research methodology and the simulation model.

3.1. Overview

In order to quantify the impact of conceptual dynamics in a domain on the coverage of current elements in an ontology, we designed a simulation model, based on a colored Petri-net. For an overview of Petri-nets and their application to simulation, see e.g. [10]. Petri-nets consist of places, transitions, directed arcs, and tokens. Dynamic behavior in Petri-nets is basically represented in a Petri-net by the consumption of input tokens and the creation of new output tokens by a transition (this process is known as “firing” of a transition). That means that the structure of the net is defined at build-time of the simulation and the dynamics is represented by a flow of tokens through the net at run-time.

Simple Petri-nets allow only one token per place; a transition will fire if all input places contain exactly one token and all output places are empty. In this case, one token from each input place will be consumed and one on each output place will be created.

There are numerous extensions of Petri-nets, in particular “colored” Petri-nets and such that model time explicitly. In colored Petri-nets, tokens can be distinguished in that each token may carry certain properties. The transitions may contain conditions that refer to the properties of tokens on the input places. When a transition fires and consumes tokens from input and creates tokens in the output, the transition may assign any kind of locally generated results to the created tokens on the output places.

For Petri-nets, there is a wealth of simulation environments available. For our experiments, we chose the commercial package PACE[1] version 4.0. PACE is based on colored Petri-nets and uses Smalltalk as the language for specifying conditions and other instructions inside the transitions.

3.2. Simulation Model

The basic idea of our model is as follows:

1. From domain data, we extract a set of past events that can be regarded as the birth of a new conceptual element, and store the day of its first appearance.
2. If possible and reasonable, we obtained or estimated the lifespan of the conceptual entity, i.e. the duration for which the element would belong to the active vocabulary. Though we do not need to remove non-conflicting, outdated elements from a domain ontology, such may be helpful because we can use it later for determining the domain coverage of an ontology as a percentage over the elements in the current state of the real world – e.g. how many concepts are included in the ontology vs. how many are used in the real world.
3. We assume that the time behavior of the ontology engineering and maintenance process can be approximated using one of the following patterns or a combination of those:
   a) **Regular update in fixed time intervals:** A new ontology version is produced at regular points in time. This new version will include all conceptual elements that are waiting in the processing queue. In order to make the model more realistic, we also introduced a lead time, i.e. elements must arrive a certain amount of days before the next release in order to be included. Those that arrive later will only be processed in the update following the upcoming one.
   b) **Fixed processing capacity per time:** A new ontology version is produced as in a), but additionally, the processing “bandwidth” for the ontology is limited in that only a maximum number of change requests can be processed in a given amount of time.
   c) **Minimum amount of change requests:** A pre-defined minimum of change request must be waiting in the queue for triggering the update of the ontology.

Since the process of adding the elements is modeled as one transition, it can be easily extended so that it reflects the time behavior in more detail. For example, multiple stages or voting and review mechanisms can be added easily.

Following that guideline, we designed the simulation model as shown in Figure 2. The place P1 will hold all tokens that represent such conceptual elements that will

[1] http://www.ibepace.com/

become relevant during the analyzed time-span. For example, we created tokens for patent applications that reflect novel procedures or materials. Transition T1 will fire once the date of birth of a token in P1 is reached. It creates two new tokens, one on P3 and one on P2. The place P3 reflects the set of conceptual elements in the real world – quite trivially, these are all tokens from P1 that have already been "born", minus those that have already become irrelevant. The latter process is represented by T5: Once the current time is greater or equal to the date of birth plus the lifespan of a conceptual element, it is removed from P3.

The place P2 reflects the maintenance queue of the ontology engineering process, i.e. such elements that have been added to the real world but are yet to be incorporated in the next ontology release. The process of adding a waiting element to the next ontology release is represented by T2. The conditions for this transition can be set to reflect the various update patterns a), b), or c) as described above. Once T2 fires, it consumes the waiting conceptual element from P2 and creates a new element on P4. The place P4 represents the current ontology release, i.e. the set of conceptual elements that have yet been added to the ontology. Same as in the real world, outdated conceptual elements are removed from the ontology via firing of T3 if the lifespan has lapsed. We did not model explicit removal and a new lag in here, since the removal of outdated conceptual elements is used only to be able to determine the domain coverage of the ontology as a percentage of the current domain elements.

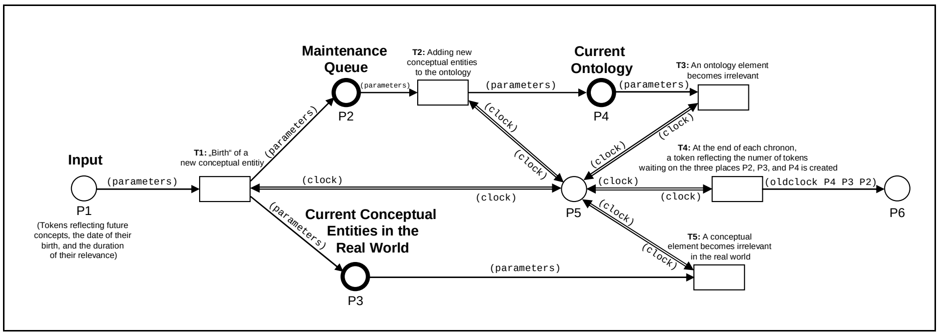
Maintenance Queue
T2: Adding new conceptual entities to the ontology
Current Ontology
T3: An ontology element becomes irrelevant
(parameters)
(parameters)
(parameters)
P2
P4
(clock)
(clock)
(parameters)
(clock)
(clock)
T4: At the end of each chronon, a token reflecting the numer of tokens waiting on the three places P2, P3, and P4 is created
T1: „Birth" of a new conceptual entitiy
Input
(parameters)
(clock)
(clock)
(oldclock P4 P3 P2)
(clock)
(clock)
P1
P5
P6
(Tokens reflecting future concepts, the date of their birth, and the duration of their relevance)
(parameters)
Current Conceptual Entities in the Real World
(clock)
T5: A conceptual element becomes irrelevant in the real world
(clock)
(parameters)
P3

Fig. 2. Petri-net model of conceptual dynamics and ontology maintenance

In the model, we represent time by discrete time tokens waiting in P5. Each action that takes place at a given moment in time consumes such a time token and creates and returns a new one immediately. This is depicted by the double-lined arcs with arrows at both ends between P5 and the respective transitions (this is a common notion in several Petri-net tools). Only if no other transition (e.g. T1, T2, T3, or T5) consumes the time token waiting on P5, transition T4 will fire and create two new tokens, one new time token with an increased time value "clock" on P5 and one token on P6. The priority of all other transitions is implemented by a delay in the definition of T4. The newly created token on P6 has as attributes the current amount of tokens waiting on P2, P3, and P4 plus the respective time. These tokens reflect the amount of conceptual elements available in the real world (P3), those already contained in the current ontology (P4) and those waiting in the ontology maintenance queue (P2) for every discrete moment in time within the simulated time-span. They can be used easily to draw time series diagrams for those values.

The “clock” property of the time tokens is stored as a positive integer value indicating the number of days from January 1, 1900 on (i.e. 1 = Jan 1, 1900; 2= Jan 2, 1900, …). Table A-1 in the annex shows the specifications of the firing conditions for the five transitions in the model. In the Smalltalk dialect used in PACE, the modulo operator is written as two backslashes. Thus, the condition that a transition shall fire if the current day in the simulation run can be evenly divided by 15 (“`clock mod 15 == 0`”) must be written as “`clock\\15=0`”).

3.3. Model Validation and Test Run

In order to validate the simulation model, we created test data covering 120 days. On each of those days, exactly one new conceptual element is being born, and the lifespan of each element was set to 30 days. The ontology maintenance lag was defined as 15 days plus 7 days lead time (i.e.: the ontology is updated every 15 days, and all elements are included that are known at least seven days before the next release).

Accordingly, the condition for T2 was set to

```
(clock\\15=0) & (clock>=((parameters at: 2)+7))
```

The resulting data from the test run is visualized in Figure 3. One can see that after a typical lead-in phase, the percentage of coverage of the ontology oscillates in a sawtooth wave. Every 15 days, all elements that have been known at least 7 days before the next update, will be added to the ontology. The amount of elements in the real world remains stable after day 30, because on each day, one new element enters the domain and one old one becomes irrelevant and leaves the active vocabulary.

We can see that the simulation model behaves as predicted and can be used for further analysis.

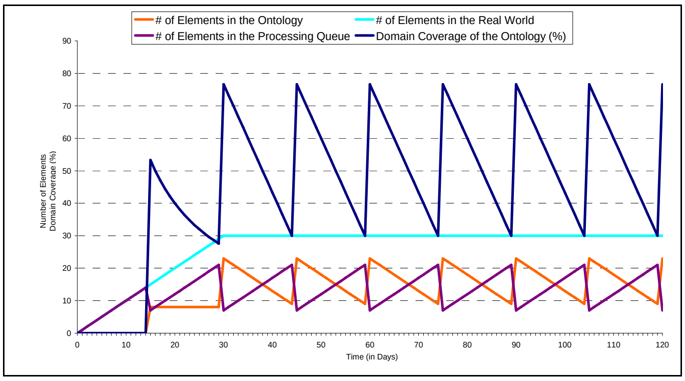

Fig. 3. Visualization of the test run

3.4. Model Calibration

For all further experiments, we assumed that a new ontology release is produced every 360 days and that it will include all waiting change requests as long as they are available at least seven days before the update. Of course, shorter update cycles may be possible in some ontology projects. A quick look on ontologies on the Web showed however that one comprehensive update per year is rather the notable exception than the rule.

Accordingly, we set the condition for T2 to:

```
(clock\\360=0) & (clock >=((parameters at: 2) + 7))
```

4. Data Sources and Pre-processing

Our simulation model expects input data as a set of triples of the form `(ID, dateOfBirth, lifeSpanInDays)`. That means that the ideal type of input data would be a log file of “birth“ dates of all conceptual elements in a clearly defined domain of discourse. Each entry should specify the date on which the element was introduced to the domain and indicate the time-span for which this element belongs to the active vocabulary. Also, the log file should cover at least multiple times the average lifespan of a single conceptual element.

Since nobody keeps such a complete register of the birth and death of conceptual elements, respective data is not readily available. Thus, we derived the input data for our experiments from reasonable proxies for the appearance of new conceptual elements. There are at least three promising types of sources for such data:

1. Public announcements of relevant concepts or ontological instances, e.g. the official release data of the first product of a new kind.
2. Patent applications, in which an entity seeks protection of the exploitation rights for a new device, method, or material.
3. Filing data or public notices of applications for public approval, e.g. such in which an entity seeks permission for a new type of treatment or the introduction of a new type of medical product.

Alternatively, one could generate random input data based on assumptions about the distribution properties. However, since such distribution properties are hard to determine and hard to justify, and since we were able to find promising real data sources, we did not consider this option.

For our simulation, we tried to find data sources for individual areas that are coherent enough to be considered a domain of discourse and for which the creation of a domain ontology would very likely provide business benefit. We were successful in obtaining such data for three selected areas, i.e. for

1. Intel CPU models as core computer components,
2. pharmaceuticals, and
3. methods, recipes, and procedures in the inorganic chemical technology sector.

4.1. Intel CPUs

Intel CPUs are important conceptual elements when describing the market for computers and computer peripherals. This is because each individual Intel CPU make and model (e.g. “Pentium III 300 MHz”) is helpful for describing the performance and characteristics of a desktop or laptop computer. For makers of motherboards or other components, being able to specify the supported types of CPUs by referring to an ontology will also provide benefits. Intel CPU makes and models are rather classes than instances, since the actual instances are the tangible, mass-produced circuits. Depending on the ontology and the ability of the ontology language to support meta-modeling, they may also be needed as instances in the ontology. Whatever the decision on this modeling choice is, we are confident that Intel CPUs are relevant conceptual elements of the computer parts domain. Also, a particular Intel CPU is more specific than just a type of CPU for which the manufacturer is "Intel" and the clock speed is e.g. 768 MHz – each CPU model is a conceptual entity of its own.

From [16], we were able to obtain a complete list of the release dates of all Intel CPU models back to the famous model 4004 released in 1971. Table 1 shows an excerpt from the respective data. For our experiments, we focus on the conceptual dynamics between January 1, 1997 and January 1, 2002.

As a speculative extension of the experiment, we estimated the "conceptual lifespan" of such CPU models, i.e. for how long they actually belong to the active vocabulary. Due to the high degree of innovation in markets, outdated CPU models are for example almost completely irrelevant when describing offers in the E-Commerce domain. We guessed that CPUs released until the end of 1997 would belong to the relevant concepts for 720 days after their introduction and those introduced from 1998 onwards for 360 days. These estimates are based on a subjective assessment of the pace of innovation on the PC market, and on the increased amount of new releases after 1998.

Of course one can argue that there will always be some old data referring to past CPU models and thus such estimates are always questionable. However, we can reasonably assume the biggest share of offerings data in the e-commerce domain to refer only to *current* CPUs. Also, the proposed extension is not necessary for the main experiment. It will, however, allow us to measure the domain coverage as a *percentage* of ontology elements over current conceptual elements in the real world (i.e.: "how much world is in the ontology").

Table 1. Sample from the Intel CPU data (based on [16])

Release Date ("Date of Birth")	Release Date (Integer Value)	Intel CPU Model
26.01.1998	35821	Pentium(r) II Processor (333 MHz)
02.04.1998	35887	Mobile Pentium(r) II Processor (233 and 266 MHz)
15.04.1998	35900	Intel(r) Celeron(r) Processor (266 MHz)
15.04.1998	35900	Pentium(r) II Processor (350 and 400 MHz)
08.06.1998	35954	Intel(r) Celeron(r) Processor (300 MHz)
29.06.1998	35975	Pentium(r) II Xeon(tm) Processor (400 MHz)

4.2. Pharmaceuticals

In the field of pharmaceuticals, it was rather difficult to obtain meaningful data. Eventually, we decided to take the filing data of applications for the approval of new pharmaceuticals in the United States as our main data source. As raw data, we used the drug application data from the US Food and Drug Administration (FDA) from

July 1996 through July 2002, which we could obtain from [17] in the form of HTML files. The FDA applications are divided in categories as follows [cf. 17]:

a) „Original New Drug Applications“,
b) „Efficacy Supplemental New Drug Applications“,
c) „Approvable Original New Drug Applications“,
d) „Original Abbreviated New Drug Applications“,
e) „Original Abbreviated and 505(b)(2) New Drug Applications with Tentative Approval“, and
f) „Labeling Supplements to Original New Drug Applications“.

For our analysis, we only considered category a), since only these are truly novel drugs. Applications in this category are described in more detail using the attribute “chemical type”. We removed all entries that are of the subtype „Already marketed drug but a new manufacturer“ (type 5, see [17], 20 entries) und two obvious redundancies (#21015, ANDROGEL) and (#21124, LAMISIL). All in all, this returns 481 valid elements.

This selection of data corresponds to an ontology of pharmaceutical substances and products. This could for example be used to support procurement processes of pharmaceuticals, annotate prescription data for mining purposes, or semantics-supported healthcare applications, etc. We retrieved all monthly reports, merged them into one large HTML file, and extracted the plain text data. Then, we used a small Java program to export the relevant data fields and write them to a CSV file. Table 2 shows a sample from the respective data.

Table 2. Example of the FDA pharmaceutical data (based on [17])

Original Application No.	Approval Date	Tradename
20616	03. Jul 96	KADIAN
20536	03. Jul 96	NICOTROL
20630	12. Jul 96	ULTIVA
50711	18. Jul 96	ZITHROMAX
20554	22. Jul 96	DOVONEX
20625	25. Jul 96	ALLEGRA

4.3. Methods, Materials, and Procedures in Inorganic Chemistry

As the third segment in our analysis, we looked at innovation in the inorganic branch of chemistry. The starting point for our data were German patent applications. We assume that patent applications are a good estimate for the lower limit of the conceptual dynamics in this domain, since the application for a patent is associated with cost and effort. It is thus safe to assume that the applicants expect business significance of the innovation – both economically and conceptually.

Since collecting the data from a Web database showed to be very labor-intensive, we constrained our analysis to the branch of *inorganic* chemistry. The source of our data was the database of the German patent and trademark registry [18]. We retrieved pending and approved applications for patents and utility models and regarded each application as a surrogate for a novel conceptual element in the domain. The system supports search by patent categories using International Patent Classification (IPC) codes. The IPC is a hierarchical classification schema for patents. We selected section

C „Chemistry; Metallurgy“, subsection C01 („Inorganic Chemistry“) [19]). Within that subsection, we considered the segments C01B, C01C and C01D. Those three are defined as given in Table 3 ([cf. 19]):

Table 3. Definition of the IPC patent categories C 01 B, C, and D

IPC	Definition
C 01 B	Non-metallic elements; compounds thereof
C 01 C	Ammonia; cyanogen; compounds thereof
C 01 D	Compounds of alkali metals, i.e. lithium, sodium, potassium, caesium, or francium

We queried the database for the period from January 4, 1999 – December 15, 2002 on a weekly basis. In order to keep the amount of queries in a reasonable order of magnitude, we used the last day of the respective week as the „birth date“ and not the actual date of the application. However, we assume that this small deviation can be neglected. All in all, we carried out 624 manual queries (52 weeks * 4 years * 3 patent categories).

We took into account all entries from the section „patent applications and utility models“, using the date of the application document as the date of birth for the respective conceptual element. All in all, we collected 490 application documents for patents and utility models for the given period of time. In this case, we manually extracted all data from the HTML files per each calendar week. We then wrote a small Smalltalk script for creating respective tokens on place P1 in the simulation environment. The simulation run started with January 10, 1999 (birth of the first element, day 36170 as an integer) and ended on December 15, 2002 (day 37605 as an integer).

5. Results

In this section, we summarize the results from the three simulation runs and discuss on how the conceptual dynamics in the selected domains influences the construction of domain ontologies. As said, we assumed one ontology update per each 360 days with a deadline for inclusion of seven days before the respective update takes place. Table 4 gives an overview of the amount of missing elements in a respective domain ontology created in such a setting.

Table 4. Amount of missing elements in the three domains

Domain	Time-span analysed	Amount of Missing Elements		
		Mean	Median	Max
Intel CPUs	1/1997 – 1/2002	6.835	5	24
Pharmaceuticals	7/1996 – 6/2002	43.689	38	129
Inorganic Chemical Innovations	1/1999 – 12/2002	49.321	45	150

We can see from the median value that during half of the time, at least five Intel CPU concepts, 38 pharmaceutical innovations, and 45 concepts reflecting new methods, materials, or procedures in the inorganic chemical industry sector, are missing in the respective ontology. Shortly before the next ontology update, the number of missing elements rises up to 24, 129, and 150 respectively. It is important to stress that those missing elements reflect the innovative part of the domain, which has usually much higher business relevance for search and information processing tasks. Not being able to use semantic technology for processing data that refers to those “hot topics” may drastically reduce the business value of semantic technology in the respective domains. For example, few people searching for a place that sells bread or butter will consult the Web, as compared to someone searching for “wakeboarding on Mauritius” or a “Bluetooth noise-cancelling headset for Nokia”. Similarly in the B2B segment, a producer of pharmaceuticals may want to use Semantic Web technology for watching all news and blog entries referring to a novel type of product of the competition, and a semantics-supported knowledge base for customer support of PC manufacturers may require that we annotate incidents using an ontology of the involved CPU model.

In the following, we present the detailed simulation results per each domain.

5.1. Intel CPU Ontology

Figure 4 shows how the amount of conceptual elements in an Intel CPU ontology will develop over time. We can see clearly that the conceptual dynamics in this domain is increasing from year to year. During at least half of the time, five or more such concepts are missing in the domain ontology (median = 5); shortly before the annual update, this rises to a maximum of 24 missing entries.

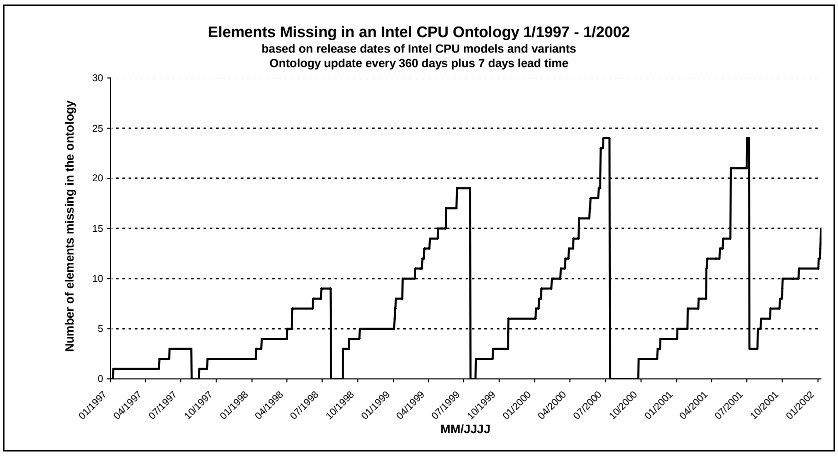

Fig. 4. Amount of missing elements in an Intel CPU ontology

In this domain and based on the assumptions described in section 4.1, we were also able to estimate the *full* size of the active vocabulary in such an ontology. If we use these lifespan estimates, then we can determine the median number of CPU model concepts in the active vocabulary as 19; the mean is 16.961. This says that about half of the time, five or more of the CPU types in an active vocabulary (of on average) 17 CPU types are not yet included in the ontology. And, unfortunately, those are likely the most interesting ones for business entities, and those for which Semantic Web-based comparison shopping would be most attractive due to high price dispersion among novel products.

We also computed the domain coverage for each day in the simulation run. For this, we divided the amount of elements in the domain ontology (i.e. the number of tokens on place P4) by the amount of elements in the real world (place P3). For the series of the resulting values, the mean is 61.55 %. In other words, on average almost 40 percent of current CPUs would be missing in the ontology!

5.2. Ontology of Pharmaceuticals

Figure 5 shows how the amount of conceptual elements in an ontology of FDA-approved pharmaceuticals will develop over time. The simulation period was from July 3, 1996 through June 26, 2002. In this domain, we see a slight decrease in the conceptual dynamics from year to year. Also, when compared to the overall amount of FDA-approved drugs, the average of 43 missing elements may be acceptable. This reflects that, due multi-stage clinical trials and in general the long time-to-market of new pharmaceuticals, the dynamics in this domain is heavily constrained by legal regulations.

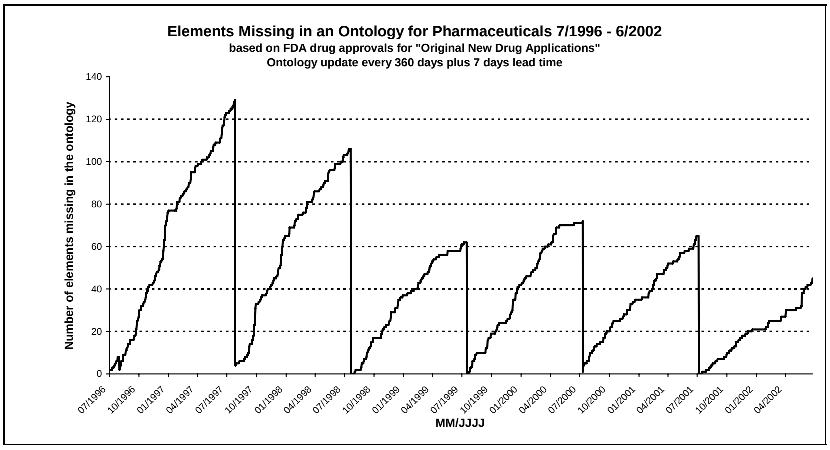

Fig. 5. Amount of missing elements in an ontology of pharmaceuticals

5.3. Ontology of Methods, Materials, and Procedures in Inorganic Chemistry

Figure 6 shows how the amount of conceptual elements in an ontology of methods, materials, and procedures in inorganic chemistry will develop over time. We can see clearly that the conceptual dynamics in this domain is increasing year by year. This may reflect either a general increase in research and development productivity, or just an increase in the tendency to seek patent protection. During at least half of the time, 45 or more such concepts are missing in the domain ontology; shortly before the annual update, this rises to a maximum of 150 missing entries.

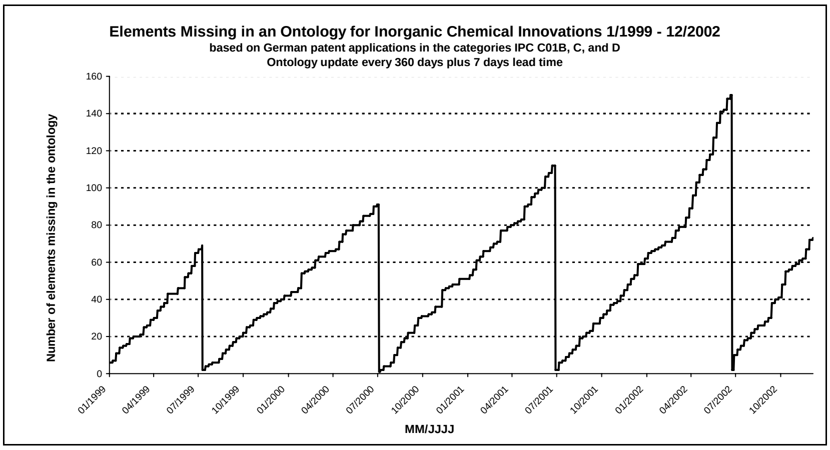

Fig. 6. Amount of missing elements in an ontology of inorganic chemical innovations (derived from patent and utility model applications)

The total number of innovations in the domain of discourse over the full duration of the experiment was 490. For reasons of comparison, we also determined how the amount of missing elements would be reduced if the ontology was updated every 21 days (still requiring seven days lead time). In that case, the mean would be just 6.74 missing elements.

Since we can assume that the majority of innovations in this domain will have a lifespan that by far exceeds the duration of our simulation run, we cannot determine the absolute size of the current vocabulary in the real world.

6. Evaluation and Discussion

In this section, we relate our research to previous work and discuss its implications.

6.1. Related Work

Steels and Kaplan carried out simulation experiments on how a group of autonomous agents can update their shared vocabulary so that it incorporates semiotic dynamics in the community of these agents [20]. Fensel stressed that building ontologies solely based on the search of ontological truth is problematic, because human actors are able to find shared domain conceptualizations only in a social process by means of perception and argument, in which the conceptualization is both means and object [1]. Thus, Fensel claims that dynamics in the domain is caused not only by factual, objective changes but also by progress in the argument and changes in the perception of the world. Currently, there is a lot of interest in re-using the Wiki-approach as an ontology engineering environment. This has in common with our work the assumption of domain dynamics as a focal point of ontology building. Our findings give additional evidence for the justification of this assumption.

Oliver et al. analyzed how change in medical terminology caused by scientific advancement can be managed in controlled vocabularies [21]. Schulze and Stauffer carried out simulation experiments on the diffusion of languages based on individual variation, passing along, and selection [22]. In contrast to our work, their analysis focuses on the dynamics of the adoption of languages as larger units, while we address dynamics at the level of individual concepts. Additionally, earlier experiments of ours have been reported in [13]. We do not know of any other quantitative research on the interplay of domain dynamics and ontology coverage.

6.2. Implications

We have provided a generic Petri-net-based model for relating the dynamics in a domain to producing and maintaining conceptual models for that domain. One of the properties of the Petri-net approach is that the temporal behavior of the ontology maintenance process can be modified or refined as needed, so that it would reflect various modes of maintenance, e.g. fix intervals, bandwidth/capacity constraints, or delays in a multi-stage process with explicit voting etc. Also, the simulation model itself can be applied to any conceptual modeling problem.

The simulation runs clearly show that there is a substantial amount of conceptual dynamics in the three domains. In the case of an Intel CPU ontology, the results are most obvious. Firstly, the benefits of a respective domain ontology are easy to identify. Secondly, based on assumptions about the duration of domain relevance of a new processor model, we could even determine the average domain coverage in percent. All in all we can see that the order of magnitude of domain dynamics poses a challenge for building current domain ontologies, and that this challenge is on top of the technical challenges addressed by available infrastructure for ontology versioning and evolution. Unfortunately, we can assume that in many application domains, the most novel concepts must be available in the ontology in order to exploit the business potential of semantic technology.

In dynamic domains, the possible degree of domain coverage is constrained in two ways: First of all, the group of individuals building the ontology must be aware of the novel conceptual elements. In here the bottleneck is often whether users of the ontology have an easy-to-use mechanism of reporting missing elements. Secondly, the lag and quality of the ontology maintenance process limits the inclusion of such change requests. This is a strong indicator that in domain ontologies, the problems of ontologies constructed by a small “elite” but meant for a bigger set of users is more problematic than for abstract upper-level ontologies [cf. 14].

Of course, we can mitigate most of the problems by increasing the level of abstraction. However, materializing most of the promises of the Semantic Web will require very detailed domain ontologies. After all, most of the data exchanged in commerce is referring to very specific categories of things – few people order „hot beverages“ in a bar (rather „a café latte with macadamia flavor“) and employment agencies do not search for „human actors“ (rather „mechanical engineer with >= 3 years of professional experience in maintaining car-wash systems“).

The only feasible approach for dealing with dynamic domains is speeding up ontology maintenance. It is obvious that monthly or weekly updates of the ontologies in our simulation experiments will drastically reduce the amount of missing elements. However, the bigger the number of stakeholders involved in the conceptualization and formalization of an ontology, the more time will be necessary for reaching agreement.

In the following, we summarize findings from our experiments:

(1) Not only the absolute duration of the update interval is relevant, but also the relative timing of the release dates. This is in particular true if the usage of the ontology will be unevenly distributed over time, e.g. due to seasonal effects.
(2) Proper conceptual modeling alone does not solve the problem of domain dynamics.
(3) The group of individuals taking care of the maintenance of an ontology must establish mechanisms that make it as simple as possible and as rewarding as possible for plain users to report change requests. Otherwise, missing elements may be spotted but never reported.
(4) We need to think of ontology modularization also in terms of decoupling domain dynamics and distributing responsibility. In this respect, ontology engineering can learn from the lessons in creating lasting numbering schemas like EAN/UPC or the ISBN, for which the standardization bodies assign authority over the subsets of the naming space according to a hierarchical schema. This allows e.g. any company in the world to define globally unique EAN/UPC codes within their branch without delay.
(5) We may be able to predict the emergence of a novel concept. In that case, already the early indicators should trigger the change request.
(6) In some cases, lightweight ontologies that capture evolving classification schemas as pure literal values may be a better solution than to continuously replicate the schema as an ontology formalization.

7. Conclusion

The reasons for the current shortage of domain ontologies are manifold, but the pressure caused by dynamics in many domains and the resulting complexity of the management of change and ontology evolution is for sure one of them. In this paper, we have presented a general simulation model that allows quantifying the conceptual dynamics in relevant domains and determining the amount of missing elements under typical ontology engineering settings. The experiments show that ontology engineering at the level of domain ontologies faces a substantial innovation dynamics in the domains, which will lead to insufficient domain coverage if the ontologies are updated only infrequently. This is a major problem since it will often be the very new conceptual elements in domain ontologies that empower semantic systems to provide business benefit in terms of agility and operational efficiency. Current approaches of bringing domain ontology engineering back into the hands of the users, e.g. on the basis of Wiki technology, are likely an important direction.

Acknowledgements: The work presented in this paper has been partly supported by the European Commission under the projects SUPER (FP6-026850) and MUSING (FP6-027097), and by the Austrian BMVIT/FFG under the FIT-IT project myOntology (Grant no. 812515/9284). Martin Hepp is also supported by a Young Researcher’s Grant (Nachwuchsförderung 2005-2006) from the Leopold-Franzens-Universität Innsbruck.

References

1. Fensel, D.: Ontologies: Dynamic networks of formally represented meaning. Available at http://sw-portal.deri.at/papers/publications/network.pdf (retrieved December 15, 2006).
2. Ventrone, V., Heiler, S.: Semantic heterogeneity as a result of domain evolution. ACM SIGMOD Record 20 (4): 16-20, 1991.
3. Noy, N.F., Klein, M.: Ontology Evolution: Not the Same as Schema Evolution. Knowledge and Information Systems 6 (4): 428-440, 2004.
4. Heflin, J., Hendler, J.: Dynamic Ontologies on the Web. Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI), Austin, Texas (2000): 443-449.
5. Klein, M., Fensel, D.: Ontology versioning for the Semantic Web. Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA (2001): 75-91.
6. Klein, M., Ding, Y., Fensel, D., Omelayenko, B.: Ontology Management: Storing, Aligning and Maintaining Ontologies. In: Davies, J., Fensel, D., Harmelen, F.v. (eds.): Towards the Semantic Web. Wiley (2003) 47-69.
7. Vrandecic, D., Pinto, S., Tempich, C., Sure, Y.: The DILIGENT knowledge process. Journal of Knowledge Management 9 (5): 85-96, 2005.
8. Pinto, H.S., Martins, J.P.: Ontologies: How can They be Built? Knowledge and Information Systems 6 (4): 441-464, 2004.
9. Hepp, M., Leukel, J., Schmitz, V.: A Quantitative Analysis of Product Categorization Standards: Content, Coverage, and Maintenance of eCl@ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary. Knowledge and Information Systems (forthcoming), 2007.
10. Peterson, J.L.: Petri Nets. ACM Computing Surveys 9 (3): 223-252, 1977.
11. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5 (2): 199-220, 1993.
12. Noy, N.F., Hafner, C.D.: The State of the Art in Ontology Design. AI Magazine 18 (3): 53-74, 1997.
13. Hepp, M.: Güterklassifikation als semantisches Standardisierungsproblem. Deutscher Universitäts-Verlag, Wiesbaden (2003).
14. Hepp, M.: Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. IEEE Internet Computing 11 (7): 96-102, 2007.
15. Uschold, M., Grüninger, M.: Ontologies: Principles, Methods, and Applications. Knowledge Engineering Review 11 (2): 93-155, 1996.
16. Intel Corp.: Intel Microprocessor Quick Reference Guide. Available at http://www.intel.com/pressroom/kits/quickrefyr.htm (retrieved December 15, 2006).
17. U.S. Food and Drug Administration: FDA Drug Approvals List. Available at http://www.fda.gov/cder/da/da.htm (retrieved December 18, 2006).
18. Deutsches Patent- und Markenamt: DPINFO. Datenbank zu Patenten, Gebrauchsmustern, Marken und Geschmacksmustern. Available at https://dpinfo.dpma.de/index.html (retrieved December 15, 2006).

19. World Intellectual Property Organization: IPC 7 English Version Section C. Available at http://www.wipo.org/classifications/fulltext/new_ipc/ipc7/ec.htm (retrieved January 10, 2003).
20. Steels, L., Kaplan, F.: Collective learning and semiotic dynamics. Proceedings of the 5th European Conference on Artificial Life (ECAL99), Lausanne, Switzerland (1999): 679-688.
21. Oliver, D.E., Shahar, Y., Shortliffe, E.H., Musen, M.: Representation of change in controlled medical terminologies. Artificial Intelligence in Medicine 15 (1): 53-76, 1999.
22. Schulze, C., Stauffer, D.: Monte Carlo simulation of the rise and the fall of languages. International Journal of Modern Physics C 16 (5): 781-787, 2005.

Annex: Details of the Simulation Model

Table A-1. Specification of the transitions in the simulation model

Tx	Condition and PACE instructions	Meaning
T1	`Clock>=(parameters at: 2)`	A new conceptual element from the simulation input data is “born” when its “date of birth” is reached. Consequently, two tokens for this new element are created; one is immediately added to the set of current conceptual entities in the real world (P3) and another one is added to the maintenance queue (P2), where it waits for being added to the current ontology.
T2	`clock\\360=0 & (clock>=((parameters at: 2)+7)`	A conceptual element in the maintenance queue (P3) is added to the current ontology (P4) if the current day is 360 days after the last update and if the conceptual element has entered the maintenance queue no later than seven days before the ontology update.
T3	`clock>= ((parameters at: 2) + parameters at: 3)`	A conceptual element is removed from the relevant part of the current ontology (P4) if the current date is equal to or greater than the date of birth plus the predetermined lifespan of this element.

T4	`oldclock:=clock.`
	`clock:=clock+1.`
	`P4:= (self placeNamed ‘CurrentOntology’) marking tokenNumber.`
	`P3:= (self placeNamed ‘RealWorld’) marking tokenNumber.`
	`P2:= (self placeNamed ‘MaintenanceQueue’) marking tokenNumber.`
	`i:=(self placeNamed ‘Input’) marking tokenNumber.`
	`i==0 ifTrue: [selfTerminate]`
		At the end of each discrete time interval (i.e. each chronon), a token reflecting the number of tokens waiting on the three places P2 (Maintenance Queue), P3 (Real World), and P4 (Current Ontology) is created. These tokens can be directly used for measuring and visualizing the number of conceptual elements at each point in time at these three places.

T5	`clock>= ((parameters at: 2) + parameters at: 3)`	A conceptual element is removed from the real world (P3) if the current date is equal to or greater than the date of birth plus the predetermined lifespan of this element.