Reverse Leibniz, and then Bend It Like Beckham: Temporal Ontology Mapping as Problem-Solving Method

Hans Akkermans
AKMC Knowledge Management BV
and
Free University Amsterdam VUA, Business Informatics Department
Amsterdam, The Netherlands
email: Hans.Akkermans@akmc.nl

ABSTRACT
I discuss and construct ontology mappings between different ontologies of time. I show how you can use them as a new method to solve significant dynamics problems, by exploiting the properties of the ontology mapping. A unique feature of a nonlinear ontology mapping I propose is that it can rigorously treat infinitesimals as strictly finite computational quantities. The approach also suggests some novel, I believe intriguing, insights into the nature of time, particularly regarding "density" and "curvature" of time. The paper provides an in-depth case study in ontology mapping, offering some evidence that ontology building, mapping, and reuse is much a substantive issue, more than a matter of generic representation language and semantic tooling.

Categories and Subject Descriptors
H.4: Information Systems Applications – *Miscellaneous.*

General Terms
Theory, Algorithms.

Keywords
Time, ontology, mapping, signal analysis, dynamic systems

INTRODUCTION
Time is a very generic upper-level ontological concept. There are many different ontologies of time [1], but the two temporal ontologies most widely used [2, 3] in science and engineering are *point-based*: continuous time and discrete time. In continuous-time systems, time is represented by a real-numbered parameter $t \in \Re$. In discrete event-based systems, time is represented by a "step" variable $S \in \aleph$, i.e. an integer.

Continuous and discrete approaches represent two very different ontological viewpoints on the same concept of time. They not only differ in appearance, but also come with radically different concepts and methods, witness the mathematical and computational analysis of continuous versus discrete systems, for which there exists a vast literature spanning several centuries (e.g., [4] and [5]).

From the computational perspective, there is the additional problem that continuous analysis is based on the notion of derivatives and infinitesimal quantities (differential calculus dating back to Leibniz's 1684 article [4]). As the computer is an inherently *discrete* machine, computer methods for *continuous* systems invariably introduce approximations that are in fact a kind of systematic error (known as discretization or truncation error [2, 3]). In this paper I pose – and solve – the problem: can we construct an ontology mapping between continuous and discrete ontologies of time, which is both mathematically rigorous and computationally adequate, *and* is able to avoid systematic error in changing from one temporal perspective to the other?

More simply: *is it possible to reformulate any given form of continuous-time dynamics in discrete time, rigorously, without any compromise or computational approximation?*

The answer to this question is yes; the ontology mapping solution outlined in this paper entails a novel method that I call the T transform. Important characteristics of this new transform method are: (1) conceptually, it is a radical departure from the traditional view and techniques regarding the relationship between continuous and discrete time; (2) it succeeds in fundamentally avoiding computer-introduced systematic error in handling differential calculus; (3) it has informational advantages, by generating certain important systems information directly that is not so easy to obtain by conventional methods; (4) it gives rise to several new and elegant discrete algorithms for systems analysis; and (5) it has an extremely wide spectrum of applications and generalizations (even beyond time).

I will go through these aspects below in brief.

"NAIVE DYNAMICS": TEMPORAL ONTOLOGIES AND THEIR MAPPING

Axiomatization of Time Ontologies

Van Benthem [1] gives a tense-logical formalization of a great variety of temporal ontologies. His axiomatization for point ontologies of time is over temporal structures consisting of a non-empty set of time points ordered by a binary precedence relation <. It contains the following *shared* axioms for discrete and continuous time:

- TRANS: time ordering is transitive.
- IRREF: the property of irreflexivity; together TRANS and IRREF model the (asymmetric) notion of the flow (or arrow or "river") of time.
- LIN: linearity, expressing that time structures have a single path (or river flow bed) without branching.
- SUCC: time has no end point (continuing succession towards the future).

The difference between discrete and continuous time comes with the choice of a final temporal axiom, *either one* of the following two options:

- DENS: infinite divisibility of time, i.e. between any two time points there is always another one.
- DISC: discreteness; time is not infinitely divisible, but has the property of "stepwise" succession.

This suffices as axiomatization of the point-based temporal ontologies I consider in this paper. Van Benthem shows that this axiomatization is syntactically complete. He also shows that it admits of several models. Thus, real-numbered time $t \in \Re$ (where the axiom DENS is implied by the stronger continuity axiom CONT) and discrete event-based integer time $S \in \aleph$ (where as we shall see S indeed can be usefully read as "step") are a specific model choice for the above ontological theories. However, these are by far the most common and useful ones in scientific practice, and that's why I stick to them.

Clearly, the above two formal temporal ontologies are rather concise and simple themselves. This turns out not to be the case for discrete-continuous ontology mappings, however. I will now proceed to show that (1) temporal ontology mappings are important general constructs with many practical applications and implications, but (2) they are not unique, as several different useful ontology mappings can be constructed.

Standard Time Ontology Mapping

Let us consider first the traditional approach to continuous-discrete temporal ontology mapping, which is entrenched in today's standard techniques for numerical analysis and simulation of system dynamics and evolution [2]. Typically, one assumes that the discrete time steps or events S = 0, 1, 2, ... are embedded in continuous time $t \in \Re$ by assuming that the *integer* time point "1" (etc.) maps onto the *real* time point "1.000..." (etc.), as depicted in Figure 1. This looks very logical and natural indeed: formally the standard time ontology mapping between continuous and discrete time is (note: both ways) given by the simple *linear* function:

$$t / \tau = S \quad \text{or} \quad t = S\tau, \text{ so that} \tag{1a}$$

$$X_S = x_{t/\tau} \,. \tag{1b}$$

This equation is the ontological explication of the standard operating procedure in conventional real mathematical-numerical analysis. Here, τ denotes the free (user-selectable) parameter known as the "stepsize" in continuous systems simulation.

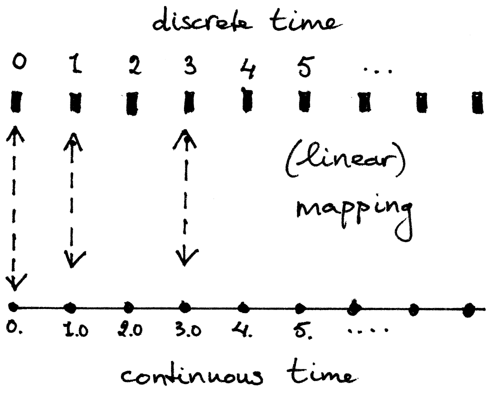

Figure 1. Traditional view on the mapping between continuous and discrete time.

How does this linear ontology mapping work in practice? Let us take a look at *the* prototypical formulation of continuous dynamic systems, viz., the ordinary differential equation[1] (ODE):

$$d/dt\, x_t = f(x_t) \tag{2}$$

As this is an equation in continuous time involving, moreover, infinitesimal calculus, it is not suitable for direct computer treatment. The standard approach then is to discretize the ODE (2) in time. The simplest choice to do so is

[1] It is conceptually interesting to reread Leibniz's original article of 1684 [4]. He clearly talks about *dx* and *dt* as *finite* differences, and then proceeds with stating the rules of differential calculus *as if* they are infinitesimals. He does not offer any justification; in actual fact his rules are incorrect for finite quantities (they neglect the higher-order differences that vanish in the infinitesimal case). He is sufficiently self-confident (or arrogant) to simply ignore this fundamental problem, and then saves the day by coming up with an important useful application example (he derives the refraction law of Snellius directly from Fermat's principle). So it seems he was lucky to live in ancient times as his article would be unlikely to survive any modern peer review. No wonder that people like the idealist philosopher Bishop Berkeley (1734) made fun out of the believers in this new calculus, deriding it as strange magic by juggling with different kinds of zeros (he added the wider point, against non-religious rationalists such as Halley, that if you do believe in this weird calculus stuff ("ghosts of departed quantities"), you also rob yourself of the right to criticize matters of theology). Proponents, among them Bernoulli, Euler, Maclaurin, D'Alembert, etc. ultimately defeated him by striking back with proper theoretical foundations of the new calculus. Interestingly, this whole conceptual struggle, involving a long string of the brightest mathematical geniuses of their time, took more than one century and a half.

by dropping the infinitesimal limit in the definition of the derivative and invoking Eq. (1b). Then

$$(x_{t+\tau} - x_t) / \tau = f(x_t) \qquad (3a)$$

or

$$\Delta X_S \equiv X_{S+1} - X_S = \tau\, f(X_S) \qquad (3b)$$

Equation (3) is computationally a one-step forward difference, generally known as the *Euler algorithm*.

The Euler formula is essentially a direct application of the standard continuous-discrete time ontology mapping of Eq. (1) and Figure 1 (where I have assumed that the step size $\tau = 1$; this simplifies the bookkeeping and can be done without loss of generality). It replaces the continuous dynamic system by a discrete-time one that is easy to compute (here, by a one-step forward recursion). It is not really used in practice, precisely because it nicely illustrates a key problem of the digital computation of continuous dynamics: it is inherently approximate (the systematic error mentioned earlier: the higher-order differences ignored by Leibniz can no longer be neglected in a finite computation).

Nevertheless, Euler's formula is rightfully seen as the grandfather of all ODE solving algorithms. Any ODE solver attempts to correct its shortcomings, lack of accuracy and sometimes also of stability, by including higher difference contributions (equivalently, orders of the Taylor expansion) up to a prespecified order. As there are zillions of ways to do this, each with specific advantages and drawbacks, this has become a computing art in itself. The famous Runge-Kutta algorithms, the universal workhorse to simulate continuous dynamic systems, are a case in point. However, this can only be done to a limited extent, as explained in a very accessible and practical way in [2]. In essence, the standard linear ontology mapping of Eq. (1) inherently and unavoidably introduces approximations in the digital computation of what basically are infinitesimal quantities.

"Naive Dynamics"

Although never stated this way, the key problem of the standard algorithms for continuous dynamics transformed to a computationally tractable discrete-time system is therefore the underlying assumption of a linear mapping between time ontologies.

With an allusion to Hayes's "Naive Physics Manifesto" (1978/85), one might say that what makes the standard linear ontology mapping attractive is that it leads to a simply understandable form of "naive dynamics" for complex (nonlinear) systems of the type (2), witness Eqs. (1) and (3). Equations (1) and (3) are both nice to have from the standpoint of naive dynamics. Unfortunately, scientific history has demonstrated that they cannot both be valid simultaneously. The standard approach then makes the choice that the linear time ontology mapping (1) is correct, but precisely this assumption invalidates the basic discrete Euler formula (3) and its descendants such as Runge-Kutta for infinitesimal calculus proper. Correcting for this is what makes the usual algorithms for continuous dynamics so complicated (or non-naive).

Now, my aim is to retain in some form this idea of naive dynamics. I will do this in a novel way, in fact the precise opposite of the standard computational approach. Specifically, I will start from the principle of the correctness of an Euler-type formula as (3). The key reason is that, if you succeed in doing this, computation and prediction of continuous systems is extremely simple, since Eq. (3) is a one-step forward difference in discrete time, and the whole future is predicted (without any approximation in the sense of built-in systematic bias, as in the standard approach) by repeated application of (3).

The necessary consequence of this alternative route is that one has to drop the correctness of the linear ontology mapping (1). However, as I will show, there is no principal reason why there can't be alternative ontology mappings with beautiful and desirable conceptual and computational properties – but consequently they must be nonlinear with respect to time. In other words, you have to "bend" time.

T: THE TRANSFORMATION OF TIME

Probabilistic Embedding of Events in Time

I now construct a new alternative class of temporal ontology mappings by means of the following procedure that embeds discrete events S in continuous time t. Imagine that the time difference (in continuous time) between the occurrence of two subsequent (discrete) events is not fixed and constant, as in the traditional approach (cf. the constant τ in Eq. (1)), but actually is *random*. So, after the start event S=0 that occurs at some given start time t_0 (taken to be t=0 in the remainder), the discrete time events S >0 occur randomly at continuous time points t_S, and the time intervals between two steps $T_1 = t_1 - t_0$, ..., $T_{S+1} = t_{S+1} - t_S$ are all random variables, governed by some given probability distribution (which I assume to be the same for all events).

Accordingly, let $P(t, S)$ be the probability that in the interval $[0,t]$ precisely S discrete events or steps have occurred. Then the time ontology mapping replacing Eq. (1) reads:

$$x_t = \sum_{S=0}^{\infty} P(t,S) \cdot X_S \qquad (4)$$

Generally, this is a *nonlinear* ontology mapping, with respect to both time variables t and S. Equation (4) actually represents a whole *class* of ontology mappings, because there are many choices for the probability function $P(t, S)$.

For this probability I now take a specific choice, namely:

$$x_t = \sum_{S=0}^{\infty} \frac{1}{S!}\left(\frac{t}{\tau}\right)^S e^{-t/\tau} X_S \qquad (5)$$

This nonlinear ontology mapping ($x_t = T(X_S)$ in short) embeds discrete events in continuous time by means of a stochastic process known as the Poisson process. Although fundamentally different from Eq. (1), it likewise has an elegant conceptual interpretation. The linear ontology mapping (1) essentially says that all discrete events occur *totally correlated* in continuous time: once we know the time instant of the initial event and the (fixed) waiting time constant τ between events, the time occurrence of all events is wholly fixed, carved in stone with military precision as it were (cf. Figure 1).

In contrast, the nonlinear ontology mapping of Eq. (5) essentially represents the opposite situation, in which all events occur *independently* and so are totally *un*correlated. This situation often occurs in reality. For example the arrival of incoming phone calls at a helpdesk is expressed by a Poisson process. Calls arrive not with fixed time intervals between them but irregularly; the probability distribution for the random time T_S between two steps is a negative exponential, and the constant τ in Eq. (5) now represents the *average* waiting time between two subsequent events.

What does this buy us? In essence, the time ontology mapping is a transform expression – the case of Eq. (5) I call the T or $\mathtt{T}$ transform – that transforms a continuous function x_t into a discrete function X_S. Transform methods are well-developed: they already stem from early 19th century mathematics, the Laplace and Fourier transforms probably the best known ones. Although mathematically demanding, their key idea is simple: if you can map the original *problem* (say, the ODE (2)) from the original space (here, continuous time) into a different problem in a new space where it is simple to solve, then you are done by simply back-transforming the found *solution* to the original space. This is what for example the Laplace transform does: it transforms differential equations from continuous time into simple-to-solve algebraic equations in frequency space. But you already find this transform idea in the solution of the mutilated chessboard problem, or in that of the children's game called Nim.

My transform idea expressed in Eq. (4) and in the T or $\mathtt{T}$ transform (5) is new and special in the sense that it transforms a problem formulated in continuous time into one that is formulated in discrete time. Discrete problems are much more suitable for solution by a computer than continuous ones; once the discrete solution is found we simply back-transform it into the continuous solution we are actually looking for by using (4) or (5).[2] That this idea practically works I am going to show now.

Key Properties of the T Transform

Some key properties of my T ($\mathtt{T}$) transform between discrete and continuous time are given in Table 1.

Table 1. Properties of the T transform Eq. (5)

Property No.	Continuous-time function $x_t = T(X_S)$	Discrete-time function $X_S = \mathtt{T}(x_t)$
I.	1 (constant)	1 (constant)
II.	t	S
III.	t^2	$S(S-1)$
IV.	t^3	$S(S-1)(S-2)$
V.	t^n	$S! / (S-n+1)!$
VI.	e^{At}	$(1 + A)^S$
VII.	$A\, x_t + B\, y_t$	$A\, X_S + B\, Y_S$
VIII.	$d/dt\, x_t$	$\Delta X_S \equiv X_{S+1} - X_S$
IX.	$d^n/dt^n\, x_t$	$\Delta^n X_S$
X.	$f_t \equiv y_t \times x_t$	$F_S = \sum_{n=0}^{n=S} [S!/((S-n)!n!)]\, \Delta^{S-n} Y_0 \times X_n$

Proofs. There are several possible derivations of the properties in Table 1, but they require some background in real mathematical analysis. Property I immediately follows from the observation that the sum of *P(t, S)* over all steps equals unity by definition, because it is a probability function. Property VII also follows immediately by direct algebraic manipulation. To prove property VIII, we differentiate both sides of Eq. (5) with respect to *t* and rearrange terms at the right-hand side, with a simple change of discrete-time variable *S*. Property IX then follows (for example) by repeating this procedure and complete induction towards the order of differentiation. Properties II-V all follow from differentiating Eq. (5) and invoking I, VIII and IX. Finally, properties VI and X are discussed in more detail in the next section in the context of various ODE applications. □

The first property (No. I) is interesting in that it conceptually implies that any constant of the motion in continuous time (think of energy, momentum, angular momentum, probability, flux) is also a constant of the motion in discrete time. The second property (No. II) states that linear functions in continuous time transform to linear functions in discrete time. These are properties that the nonlinear ontology mapping (5) shares with the linear one of Eq. (1).

[2] For the real connoisseur, I mention in passing that the linear ontology mapping of Eq. (1) can be interpreted, like my T transform (5), as a special case of the probabilistic transform (4). It is the limiting case in which the waiting-time distribution between events is the Dirac delta function $\delta(t-\tau)$. Reworking Eq. (4) on this basis by using its Laplace transform, one is led to a generating function method known as the *z* transform, which in digital control theory is also sometimes called the discrete Laplace transform. It directly yields the linear ontology mapping of Eq. (1b). Hence – although this is hardly ever explicitly recognized – also the standard numerical approach using the linear ontology mapping Eq. (1) is fundamentally based on a transform idea.

Other properties are unique to my T transform (5). In particular, continuous-time functions map onto similar (e.g. same-order polynomials, cf. properties III-VI) but not identical functions in discrete time. This is in stark contrast to the assumption in the standard received view that employs the *same* function in both continuous and discrete time (cf. Eq. (3a)). Property VII says that the T transform is a linear transform[3].

Properties VIII and IX are the crucial ones: the T transform maps the derivative d/dt onto a *finite* first-order discrete forward difference Δ. Hence, Euler-type formulas similar to (3) will be correct under the T transform. Moreover, this extends to the higher-order derivatives, which are simply found by repeated application of the finite forward difference Δ. As a consequence, a beautiful and important property of the nonlinear time ontology mapping (5) is that it in discrete time produces the higher-order derivatives of continuous time, one by one and exactly. The production of the X_S values in discrete time yields a tableau (see Figure 2), by simple subtraction or addition, that contains all desired information.

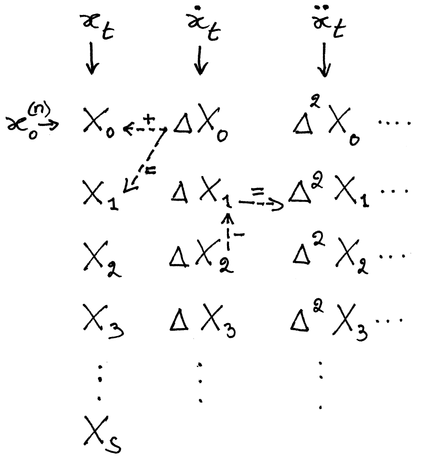

Figure 2. The T transform yields a tableau that contains the discrete solution to derivatives of any order.

If for example x_t denotes the position in a space at a certain time, the tableau not only gives the solution for the location (coefficients X_S) but it *simultaneously* solves the question as to its velocity (ΔX_S), acceleration ($\Delta^2 X_S$), etc. In addition it is able to reconstruct, by using only the T transform equation (5), the values of the continuous variables at *any* desired point in continuous time *t*. These are all major informational and computational advantages that show the power of the T transform temporal ontology mapping.

Leibniz Reversed

Consequently, we have achieved the earlier stated aim of "naive dynamics" by showing the validity of the one-step forward difference formula (3b) as the correct expression for differentiation of a continuous variable. In a sense, we have achieved this by conceptually reversing Leibniz. Leibniz talked about infinitesimals as finite quantities, and subsequently invented the correct rules of differential calculus. We took differential calculus, and subsequently invented a temporal ontology mapping that makes it actually *correct* to treat infinitesimals as finite quantities, just by switching from continuous to discrete time!

A FEW APPLICATIONS AND IMPLICATIONS

Computer Right, Man Wrong (Save Euler)

I first show how you can solve large-scale linear differential systems by temporal ontology mapping. This turns out to have an interesting side implication on the conceptual interpretation of what algorithms do.

A widely used special case of the ODE (2) is the linear system:

$$d/dt\, x_t = A\, x_t \qquad (6)$$

This equation also describes dynamic systems in many dimensions; then, *A* is not to be interpreted as a one-dimensional constant (scalar) but as a matrix. The derivation below is then generally valid for any number of dimensions.

To solve this by ontology mapping, we first transform the problem from continuous time to discrete time. Using property VIII of Table 1, the discrete version of Eq. (6) is:

$$\Delta X_S = A\, X_S \qquad (7)$$

Next, we construct the solution in discrete time starting from the known initial condition $x_0 = X_0$, and repeatedly applying the forward difference definition of the operator Δ. In effect, the whole discrete solution is stepwise produced (also in many dimensions) by the Euler algorithm (3b). From Eq. (7) it is easy to see that the discrete solution is:

$$X_{S+1} = (1 + A)\, X_S \;\Rightarrow\; X_S = (1 + A)^S\, X_0 \qquad (8)$$

Finally, this solution is back-transformed to continuous time by using Eq. (5). In the general case this can be done computationally by various methods (e.g. by successive sequences of one-step recursions or, parallelized, by matrix methods), where as a bonus you have a free choice for the time points *t* you are actually interested in. In the present case, the continuous solution is just a matter of simple table look-up, see property VI in Table 1. Hence, the ontology mapping method solves the dynamic problem (6) by first transforming the problem to a new (discrete) space, next solve it there, and then transform this solution back to the original (continuous) space, where it reads:

$$x_t = e^{At}\, x_0 \qquad (9)$$

So, we have solved a problem involving infinitesimal calculus in a strictly discrete fashion, not by directly attacking

[3] Perhaps this sounds a bit confusing, but linearity is only a *relative* notion. As stated earlier, the T transform is nonlinear with respect to the time variables *t* and *S* (see Eq. (5)). With respect to the temporal functions X_S and x_t, however, it is linear, in accordance with property VII.

the computation of derivatives in an approximate fashion (which is the standard way of doing it), but indirectly by changing the problem space first. I mention in passing that the above also yields a proof of property VI in Table 1: just insert Eq. (8) into Eq. (5) and carry out the summation.

The case discussed here has several important general applications. For example, it applies to both random walks and master equations; both have many practical applications in many different disciplines. As a bonus, the T transform proves that they are their mutual discrete and continuous-time equivalents, see also [6].

A possibly even broader application is that it is applicable to the modern state-space approach to control systems theory: adding a control signal term to Eq. (6), i.e. a function explicitly dependent on *t*, yields the fundamental systems formulation underlying control engineering of multi-dimensional continuous systems. The methods developed in this paper open up the opportunity to treat such systems by strictly discrete computer methods.

The above results give some (I believe entertaining) rehabilitation of the Euler algorithm. Let me quote a statement from [2], a remark that is prototypical for any modern textbook treatment of numerical methods: “There are several reasons that Euler’s method is not recommended for practical use, among them, (i) the method is not very accurate (...), and (ii) neither is it very stable” ([2], p. 704). In conflict with this statement, the dynamic problem (6) *has* been solved here exactly by the Euler method. However, you should interpret the results of the algorithm not as rough *direct* estimates of the continuous-time point solution (see Eq. (3a), the standard interpretation). Instead, it is to be seen as an *indirect* method producing the exact solution, however, in discrete time (according to Eq. (3b)). In conclusion, (1) evasive maneuvers do solve problems, and (2) the computer got it all right all these years, but man’s conceptual interpretation of its outputs has always been wrong (except for Euler, of course).

Shoham’s Extended Prediction Problem Does Not Exist

In his book “Reasoning about Change” (1988), Shoham worries that the usual differential dynamics (cf. Eqs. (2) and (6)) only gives a prediction of an infinitesimally small time step forward from the considered current time point *t*. So how is it actually possible at all to make predictions over extended and finite periods of time on this basis? He calls this the extended prediction problem. My ontology mapping gives a direct solution to this: it turns the derivative into a strictly discrete and finite one-step forward difference into the future. Once you have solved this finite and discrete problem, you simply transform its solution back for any desired time *t* using the T transform (5). The extended prediction problem thus seems to satisfy the quoted Bishop Berkeley 1734 characterization concerning “ghosts of departed quantities”.

Nonlinearity and the Curvature of Time: Bend It Like Beckham

The next important step is to show that the T transform method also handles *nonlinear* dynamics well. This gives it a major advantage over other transforms such as the Laplace and *z* ones. I will give a basic example of this, by considering a special case of the ODE (2), namely:

$$d/dt\, x_t = A\, x_t\, (1 - x_t) \qquad (10)$$

which is generally known as the logistic equation.

Logistic equation models. Varieties of it are widely used in practice, for example in population models of competing species in ecology. In one dimension, the linear term represents exponential growth, but the nonlinear (quadratic) term models self-limiting effects: lambs eat grass, but if there are too many in a territory (outside paradise), their population growth is ultimately restricted due to resource limitations. In more dimensions, the logistic equation can model interactions between species: lions eat lambs, but if they eat too many, first the number of available lambs will drop, and ultimately their own population numbers will go down. It is easy to imagine that such models often lead to (nonlinear) oscillatory cycles in population growth, with time delays between those of interacting species.

Solving the nonlinear logistic system (10) follows the same transform procedure as discussed above. Now, however, we have to use property X of Table 1 for its time transformation. This property might seem mathematically complex, but it is actually a discrete convolution that is computationally very simple to handle (it’s just a sequence of basic additions and multiplications). Property X can be formally proven by (rather tedious) algebraic manipulation, properly rearranging terms at the right-hand side (a much more elegant derivation uses symbolic operator algebra, but this is beyond the space of this article). This results in an analytical solution of the nonlinear ODE (10) *in discrete time*:

$$X_{S+1} = (1+A)X_S - A\sum_{n=0}^{S}\binom{S}{n}\Delta^{S-n}X_0 \cdot X_n \,. \quad (11)$$

The first term of this solution gives the linear part (as discussed above), and the second term yields the nonlinear effects. Again a variant of the Euler-type algorithm is suited to the task of prediction: from Eq. (11) it is easy to see that the discrete solution X_S obtains by successive single-step forward recursions starting from the known initial condition X_0 and then going forward in time: *S*=1, next *S*=2 etc. The probabilistic T map (5) then delivers the solution in continuous time for any desired time point *t*.

It is instructive to compare the solution (11) of the continuous ODE (10) with (i) the discrete solution (8) of the linear system (6), and with (ii) the nonlinear discrete dynamic system that is usually seen as its discrete analog (and therefore is known as the logistic map):

$$X_{S+1} = A\, X_S\,(1 - X_S) \qquad (12)$$

This logistic map is famous because it is more or less the simplest system that exhibits *chaotic* dynamic behaviour (in contrast to the logistic ODE). Again, there is a linear term and a quadratic nonlinearity, now in discrete time. But there is an essential structural difference between the discrete solution (11) to the logistic ODE on the one hand, and the linear system (8) and logistic map (12) on the other hand. The latter are iterated maps, i.e., result from repeated function application; to obtain the value at the next timepoint one only needs the preceding timepoint.

In contrast, Eq. (11) shows that in the solution of the logistic ODE all previous time points are involved. So, this continuous dynamic system has a memory in discrete time, even though this is not at all evident from the ODE formulation (10) that involves a single continuous timepoint. Although they share the name, the logistic ODE and the logistic map are totally different in their dynamic behaviour.

Lorenz chaos. Property X of Table 1 also enables to solve in discrete time the well-known Lorenz model (1963), developed to better understand atmospheric dynamics for long-range weather prediction. It became prominent because it was the first demonstration of the occurrence of chaotic behaviour in deterministic systems, with a so-called strange attractor (the famous "butterfly" shape to which the system tends in phase space). The Lorenz model is a simple 3D system with quadratic-type (in fact, bilinear) nonlinearities. So, property X directly applies, and the discrete solution of the Lorenz model has the same structure as Eq. (11).

In general, nonlinearity in continuous dynamics has the effect that it "bends like Beckham" the solution in discrete time: in contrast to the linear system Eq. (8), *all* values at time points before *S* play an explicit role in the full solution at time *S*, although the solution itself can always be computed by a one-step forward algorithm that also maintains the normal causal order of events, both in continuous and discrete time.

Preview of Coming Attractions

It is probably most interesting here to briefly investigate the *impact on the structure of time* resulting from nonlinear dynamics. Namely, the above methods and results suggest some intriguing conceptual (or if you wish, philosophical) insights into the nature of time, particularly regarding "density" and "curvature" of time.

Consider the nonlinear differential equation (2) in general and how it changes under the T temporal ontology mapping (5). The left-hand side T(lhs) is easy: according to property VIII it always maps onto a simple one-step forward difference. The right-hand side involves a composite function f(x(t)) that is generally nonlinear. Taking the transform T(rhs) changes our ontological view on the dynamic world in two stages:

- First, it changes the function f, seen as a function of x only, into a similar but not identical function F (witness for example the properties III-VI and X). This is already an important difference with the standard approach depicted in Figure 1.
- Second, it also changes the function x seen as a function of *t* (since the same properties apply again).

If we attempt to visualize this latter effect, we get a picture radically different from Figure 1. What happens is that the average *density* of the occurrence of (discrete) events is not constant but *changes* over the (continuous) time axis.

You might visualize this by imagining that the discrete time axis gets curved, and in continuous time you only see its projection onto the continuous time axis (see Figure 3). It is actually not difficult to find examples where the discrete-time "curvature" becomes so strong that it creates a singularity in (note: finite) continuous time.

Thus, the metaphor of the flow of time as a river [1] gets strangely bent due to nonlinearities: it's possible to create something like a black hole in the river bed of the timeline!

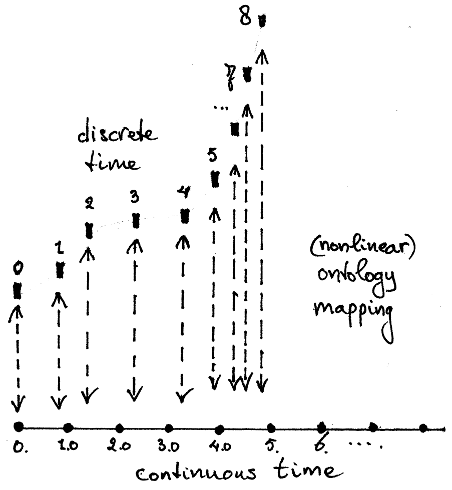

Figure 3. The T temporal ontology mapping may lead to flows of time that are "curved".

UPPER-LEVEL GENERALIZATIONS

This paper only outlines a small fraction of the results I have developed concerning nonlinear ontology mappings between time, and could only hint at the underlying mathematical proofs and algorithms. A few final general remarks are in order.

The uses of "old" science. First, the whole theory of temporal ontology mapping can be founded upon various treasures stemming from rather ancient mathematics. Much of it has more or less become extinct and superseded by modern computer (in fact, number crunching) approaches, and as a result is not treated anymore in modern textbooks on numerical methods. Specifically, this theory can be set up in a very elegant and concise way by means of symbolic operator algebra [3] as you find it in the textbook by Boole

[5] (the first edition was from 1860, by the way). What I actually find gratifying is that these methods are not just ancient, but if you develop them further as I tried to do in this paper, they can be actually made ready for today's intelligent system-style computing, and made valuable beyond popular number crunching styles of computing. A few examples have been given in this paper.

Computational complexity. An issue not discussed in this paper is the computational complexity of the algorithms related to the T transform. Generally, they are of low-degree polynomial complexity. The calculation of the T transform (5) itself is of order $O(N^2)$, where N is the number of timepoints considered (same order as matrix multiplication; this also applies to the inverse transform, as T turns out to be a so-called orthogonal transformation). The computational complexity of the solution of ODEs is commonly measured in terms of the number of function evaluations (the right-hand side of Eq. (2)). Then, the T transform method of solution is of *linear* complexity in time, cf. Eq. (11); I note that this is also true in the general case.

AI temporal reasoning and android epistemology. The present work has important differences with respect to much of the work in AI temporal reasoning (see e.g. [7]) in terms of focus and assumptions. The present work uses standard point algebra from mathematics, and therefore interval algebras and axiomatizations (such as Allen's, see also [1] and [7]) are not really relevant here. Important in my approach is that time is a *metric space*, i.e. a distance measure can be defined (in AI temporal reasoning usually called duration information). Another important difference is the type of tasks considered. AI temporal reasoning has spent much effort on constraint-based algorithms for establishing (partial) temporal ordering, possibly under incomplete or uncertain information. In contrast, this paper assumes full linear ordering in time (this is precisely what the variable S expresses), and focuses on tasks of prediction and control (in line with physics and mathematics). This paper shows that also in the point approach to temporal reasoning a lot of interesting progress can still be made.

If one refrains from delving into the mathematics, it yields some conceptual consequences for "android epistemology". Androids (computers, robots and other discrete machines such as presumably StarTrek's Mr. Data) live in a discrete spacetime. Humanoids seem to live in a continuous spacetime. So they inhabit ontologically speaking fundamentally different worlds. One might think that continuous beings can do all kinds of things in their spacetime that discrete beings cannot do in theirs – since continuous spacetime has many more points one can do something in or at than discrete spacetime. This paper shows this is not true: if they are sufficiently intelligent, discrete beings can do anything continuous beings can. Being "intelligent" can even be mathematically expressed here: the reasoning assumption that spacetime has characteristics of randomness, and still is causally ordered, according to Eq. (5).

Above I discussed things from the perspective of time. Surely this is a key top-level ontology concept. However, my approach and methods also work for types of independent variables other than time. Other continuous variables can be formally discretized in this way as well. For example, one can in this manner also treat the concept of space.

Ontology: content vs. representation. Finally, the paper has provided an in-depth case study in ontology mapping. I submit that this provides some evidence that ontology building, mapping, and reuse is much a substantive issue, more than a matter of generic representation language and semantic tooling. I note that this is already the case for such a high-level, generic, common, and commonsensical concept as time that does not depend on a specific domain. Substantive or content issues will be even more strongly present in task and domain specific ontologies. But in the end this is where the real semantic and web intelligence applications will be. This is perhaps a sign that the semantic research community at some point cannot avoid significant substantive issues in Web ontology, and has to be careful about (over)emphasis of generic representation and tooling issues without adequate domain grounding. Or, be sufficiently moderate in its expectations of the size of its own role in building the Semantic Web.

Acknowledgment. This work has been supported in part by a travel grant from the KnowledgeWeb Network of Excellence (EU-IST-2004-507482).

REFERENCES

[1] Van Benthem, J.F.A.K., *The Logic of Time*, 2nd ed., Kluwer, Dordrecht, NL (1991).

[2] Press, W. H., et al., *Numerical Recipes – The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, UK (1992).

[3] Dahlquist, G., and Björck, Å., *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ (1974).

[4] Leibniz, G.W., *Nova Methodus Pro Maximis et Minimis*, Acta Eruditorum, Vol. 3, pp. 467-473 (1684); English transl. Struik, D.J., Ed., *A Source Book in Mathematics*, pp. 272-280, Princeton, NJ (1986).

[5] Boole, G., *A Treatise on the Calculus of Finite Differences*, 2nd ed., Macmillan and Company, UK, (1872); reprint Dover, New York, NY (1960).

[6] Akkermans, J.M., and Běták, E., *Annals of Physics*, Vol. 194, pp. 148-172 (1989).

[7] Hayes, P., *A Catalog of Temporal Theories*, Technical Report (1995), http://www.ihmc.us/users/phayes/.