A Semiqualitative Methodology for Reasoning about Dynamic Systems

J. A. Ortega, R. M. Gasca and **M. Toro**
Departamento de Lenguajes y Sistemas Informáticos, University of Seville
Avda. Reina Mercedes s/n, 41012-Sevilla (Spain)

{jaortega,gasca,mtoro}@lsi.us.es

Abstract

A new methodology is proposed in this paper in order to study semiqualitative models of dynamic systems. It is also described a formalism to incorporate qualitative information into these models. This qualitative information may be composed of: qualitative operators, envelope functions, qualitative labels and qualitative continuous functions.

This methodology allows us to study all the states of a dynamic system: the stationary and the transient states. It also allows us to obtain behaviours patterns of semiqualitative dynamic systems. The main idea of the methodology follows: *a semiqualitative model is transformed into a family of quantitative models. Every quantitative model has a different quantitative behaviour, however they may have similar qualitative behaviours.*

A semiqualitative model is transformated into a set of quantitative models. The simulation of every quantitative model generates a trajectory in the phase space. A database is obtained with these quantitative behaviours. It is proposed a language to carry out queries about the qualitative properties of this database of trajectories. This language is also intended to classify the different qualitative behaviours of our model. This classification helps us to describe the semiqualitative behaviour of a system by means of hierarchical rules obtained by means of machine learning.

The completeness property is characterized by statistical means. A theoretical study about the reliability of the obtained conclusions is presented.

The methodology is applied to a logistic growth model with a delay.

Introduction

In science and engineering, knowledge about dynamic systems may be quantitative, qualitative, and semiqualitative. When these models are studied all this knowledge should be taken into account. In the literature, different levels of numeric abstraction have been considered. They may be: purely qualitative (Kuipers 1994), semiqualitative (Kay 1996), (Ortega, Gasca and Toro 1998a) and (Berleant and Kuipers 1997), numeric interval (Vescovi, Farquhar and Iwasaki 1995) and (Corliss 1995), and quantitative.

Different approximations have been developed in the literature when qualitative knowledge is taken into account: distributions of probability, transformation of non-linear to piecewise linear relationships, Monte-Carlo method, constraint logic programming (Hickey 1994), fuzzy sets (Bonarini and Bontempi 1994), causal relations (Bousson and Travé-Massuyès 1994), and combination of all levels of qualitative and quantitative abstraction (Kay 1996), (Ortega, Gasca and Toro 1998b) and (Gasca 1998).

In this paper, qualitative knowledge of dynamic systems may be qualitative operators, envelope functions, qualitative labels and qualitative continuous functions.

The paper is organised as follows: firstly, the proposed methodology is explained, that is, the transformation techniques of a semiqualitative model into a family of quantitative models, and the stochastic methods applied to obtain a database of quantitative trajectories. Secondly, the kind of qualitative knowledge we are using is introduced and the concept of semiqualitative model is defined. Thirdly, the query/classification language on this database is described. The language allows us to classify this database, and in such case, a labeled database is obtained. Machine learning algorithms are applied in order to obtain the different qualitative behaviours of the system. This behaviour is expressed by means of a set of hierarchical qualitative rules. In forthcoming papers, these machine learning algorithms will be describe in detail. Finally, a theoretical study about the reliability of the obtained conclusions is presented. This methodology is applied to a logistic growth model with a delay.

Proposed Methodology

There is enough bibliography that studies stationary states, however, the study of transient states is also necessary. For example, it is very important in production industrial systems in order to carry out optimizations about their efficiency. Stationary and transient states of a semiqualitative dynamic system may be studied with the proposed methodology. It is shown in figure 1.

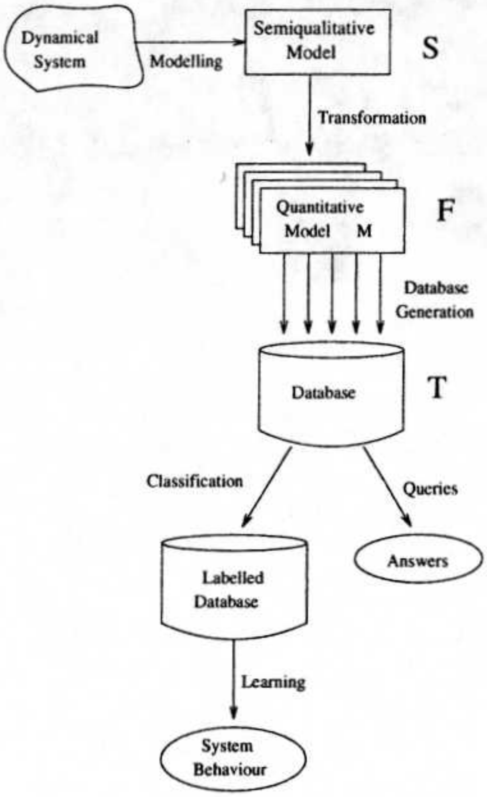

Figure 1: Proposed methodology

Let S be the semiqualitative model obtained from a dynamic system with qualitative knowledge. A family of quantitative models F is obtained from S by applying some transformation techniques. The qualitative knowledge and its transformation techniques are described below.

Stochastic techniques are applied to choose every quantitative model $M \in F$. This model M is quantitatively simulated obtaining a trajectory. A trajectory contains the values of the parameters and the values of all variables from their initial value until their final value. Therefore, every trajectory stores the values of the transient and the stationary states of M.

These quantitative trajectories are stored into a database. We propose a language to carry out queries about the qualitative properties of the set of trajectories included in the database. A labeled database is obtained when these trajectories are classified according to some criteria. Qualitative behaviours patterns of the system may be automatically obtained from this database by applying machine learning based on genetic algorithms. This algorithms are described in (Aguilar, Riquelme and Toro 1998).

In the following sections, we are going to describe the steps of the proposed methodology in detail.

Semiqualitative models

In this paper, we focus our attention on those dynamic systems where there may be qualitative knowledge in their parameters, initial conditions and/or vector field. They constitute the semiqualitative differential equations of the system.

A semiqualitative model may be composed by qualitative knowledge, arithmetic and relational operators, intervals, predefined functions, $(ln, exp, sen, \ldots)$ and numeric literals.

A semiqualitative model S is represented by means of

$$\Phi(\dot{x}, x, q, t), \qquad x(t_0) = x_0, \qquad \Phi_0(q, x_0) \quad (1)$$

being $x \in I\!R^n$ the state variables, q the parameters, t the time, $\dot{x}$ the derivative of the state variables with repect to the time, Φ the interval constraints among $\dot{x}, x, q, t$, and Φ_0 constraints with the initial conditions.

The equations (1) are transformed into a set of constraints among variables, parameters and intervals when the proposed methodology is applied. The interest of this paper is focused on those systems that can be expressed as (2) after the transformation rules have been applied

$$\dot{x} = f(x, p, t), \quad x(t_0) = x_0, \quad p \in I_p, \quad x_0 \in I_0 \quad (2)$$

where p includes the parameters of the system and new parameters obtained from the transformation rules, f is a function, and I_p, I_0 are real intervals. Equation (2) is a family F of dynamic systems depending on p and x_0.

Qualitative knowledge

Qualitative knowledge about a model may be composed of qualitative operators, qualitative labels, envelope functions and qualitative continuous functions. In this section, the representation and transformation techniques of this qualitative knowledge are described.

The representation of the qualitative knowledge is carried out by means of operators which have associated real intervals. It simplifies the integration of qualitative and quantitative knowledge, and it also facilitates the incorporation of expert knowledge in the definition of the range of qualitative variables and parameters (Gasca 1998).

Qualitative operators

Qualitative parameters and initial conditions are represented by means of qualitative operators. Every qualitative operator *op* is defined by means of an interval I_{op} which is supplied by the experts.

Unary qualitative operators Every magnitude of the problem with qualitative knowledge has its own unary operators defined.

Let U_x be the unary operators for a variable x, i. e., $U_x = \{VN_x, MN_x, LN_x, AP0_x, LP_x, MP_x, VP_x\}$. They

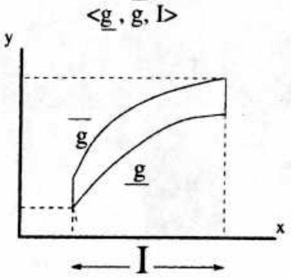

Figure 2: Envelope function

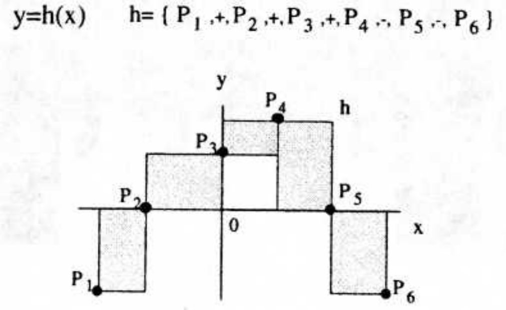

Figure 3: Qualitative continuous function

denote for x its qualitative labels: *very negative, moderately negative, slightly negative, approximately zero, slightly positive, moderately positive,* and *very positive* respectively.

The transformation rule for a unary operator is

$$op_u(e) \equiv \begin{cases} e - r = 0 \\ r \in I_u \end{cases} \tag{3}$$

being r a new generated variable, and I_u the interval associated with operator op_u which is established in accordance with (Travé-Massuyés, Dague and Guerrin 1997).

Binary qualitative operators Let e_1, e_2 be two arithmetic expressions. A binary qualitative operator $b(e_1, e_2)$ denotes the qualitative order relationship between e_1 and e_2. These operators are classified into

•Operators related to the difference $\geq, =, \leq$. The following transformation rules are applied

$e_1 = e_2$	$\equiv$	$e_1 - e_2 = 0$
$e_1 \leq e_2$	$\equiv$	$\begin{cases} e_1 - e_2 - r = 0 \\ r \in [-\infty, 0] \end{cases}$
$e_1 \geq e_2$	$\equiv$	$\begin{cases} e_1 - e_2 - r = 0 \\ r \in [0, \infty] \end{cases}$

Table 1: Transformation rules

•Operators related to the quotient $\ll, -<, \sim, \approx, \gg$ $, Vo, Ne, ...$ The applied transformation rule is

$$op_b(e_1, e_2) \equiv \begin{cases} e_1 - e_2 * r = 0 \\ r \in I_b \end{cases} \tag{4}$$

being r a new variable and I_b the interval associated to op_b in accordance with (Travé-Massuyés, Dague and Guerrin 1997).

Envelope Functions

An envelope function $y = g(x)$ (figure 2) represents the family of functions included between two defined real functions: a upper one $\overline{g} : \mathbb{R} \rightarrow \mathbb{R}$ and a lower one $\underline{g} : \mathbb{R} \rightarrow \mathbb{R}$. An envelope function is represented by means of

$$y = g(x) \ \langle \underline{g}(x), \overline{g}(x), I \rangle, \quad \forall x \in I : \underline{g}(x) \leq \overline{g}(x) \tag{5}$$

being I the definition domain of g, and x the independent variable.

The transformation rule applied to (5) is

$$g(x) = \alpha \underline{g}(x) + (1 - \alpha)\overline{g}(x) \qquad \text{with} \quad \alpha \in [0, 1] \tag{6}$$

where α is a new variable. If $\alpha = 0 \Rightarrow g(x) = \overline{g}(x)$ and if $\alpha = 1 \Rightarrow g(x) = \underline{g}(x)$. Any other value of α in (0,1) stands for any included value between $\underline{g}(x)$ and $\overline{g}(x)$.

Qualitative continuous functions

A qualitative continuous function $y = h(x)$ (figure 3) represents a constraint involving the values of y and x according to the properties of h. It is denoted by means of

$$y = h(x), \quad h \equiv \{P_1, s_1, P_2, ...s_{k-1}, P_k\} \tag{7}$$

being P_i the points of the function. Every P_i is defined by a couple (d_i, e_i), being d_i and e_i the qualitative landmarks associated to the variables x and y respectively. These points are separated by the sign s_i of the derivative in the interval between two consecutive points. A monotonous qualitative function is a particular case of these functions where the sign is always the same $s_1 = ... = s_{k-1}$.

The qualitative interpretation (figure 4.a) for every $P_i = (d_i, e_i)$ of $y = h(x)$ is

$$\begin{cases} \text{if } x = d_i \Rightarrow y = e_i \\ \text{if } d_i < x < d_{i+1} \Rightarrow \begin{cases} s_i = + \Rightarrow e_i < y < e_{i+1} \\ s_i = - \Rightarrow e_i > y > e_{i+1} \\ s_i = 0 \Rightarrow y = e_i \end{cases} \end{cases} \tag{8}$$

The transformation rules of a qualitative continuous function are applied in three steps: normalization, extension and transformation.

1. **Normalization**

   This step is used to complete and to homogenize the definition of any function.

   A qualitatitive continuous function is continuous by definition. Therefore, it verifies the following properties:

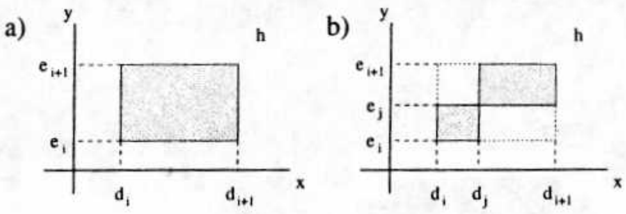

Figure 4: Qualitative interpretation of a function

- *a function that changes its sign between two consecutive landmarks passes through a landmark whose image by the function is zero,*
- *a function whose derivative changes its sign between two consecutive landmarks passes through a landmark whose derivative is zero.*

Using these properties, the definition of any function (7) is always enriched with:

- the points cutting the axes,
- the points for which the sign of the derivative changes, that is, a maximum or a minimum of h, and
- the extreme points ($+\infty$, $-\infty$), therefore, for any qualitative function (7) the first landmark is $d_1 = -\infty$ and the last is $d_k = +\infty$.

The new landmarks keep an order relationship with the old ones.

2. **Extension** The definition of any function (7) is also enriched by means of an automatic process which incorporates new landmarks. This step is carried out to diminish the uncertainty of the function since the area of the rectangle is reduced (figure 4.b).

   Let k be the number of points of a qualitative function h, if a new point is included between each two consecutive points of h, then $2k-1$ points are obtained for h. If this process is repeated i times, then the number of points that define a qualitative function h is

$$2^i k - \sum_{j=0}^{i-1} 2^j \tag{9}$$

   This number is important for the algorithm **Choose H** that is explained below.

3. **Transformation** Steps 1 and 2 complete and enrich the definition of a qualitative function. This third step obtains the representation of h as a set of quantitative functions.

   A *segment* of a qualitative function h is a sequence of consecutive points $\{P_m, ..., P_n\}$ of h separated by those points whose landmark $e_i = 0$ or where $s_{i-1} \neq s_i$. The segments divide a function into monotonous regions in which landmarks e_i have the same sign.

   A qualitative function h is transformed into a set of quantitative functions H whose behaviours are in accordance with the definition of h. The **Choose H** algorithm is applied to obtain H. It divides h into its segments and it applies stochastic techniques to choose quantitative functions that are included in H. The applied techniques are similar to Monte-Carlo method, however, they are guided to satisfy the constraints of h. The following heuristic applies a random uniform distribution to obtain the values for every landmark. First, the values that separate the segments are selected. Next, the values in every segment are selected. The obtained values must verify the order relationship among the landmarks. Therefore, they constitute a sorted sequence of numbers. There are two examples of transformation of qualitative functions in figure 5.

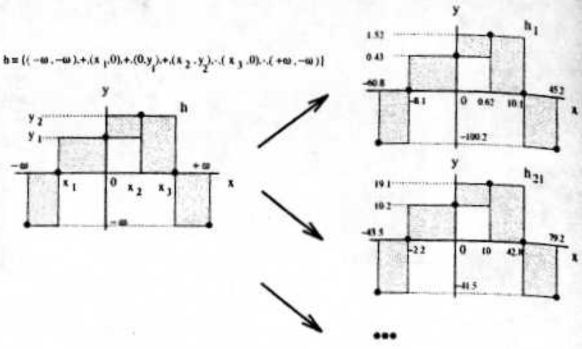

Figure 5: Qualitative function transformation

Generation of trajectories database

The aim of this section is to show the way we obtain the trajectories database T of a semiqualitative model S. A family of quantitative models F is obtained when the transformation rules are applied to S. It depends on a set of interval parameters and qualitative variables p and functions H defined by means of quantitative points.

Every trajectory $r \in T$ is obtained by simulating each particular quantitative model M which has been selected by means of stochastic techniques. In this paper, numeric simulation is carried out by means of Runge-Kutta method with variable step.

The applied algorithm is

Database generation T
  $T := \{\}$
  for i=1 to N
    $M := ChooseModel$(F)
    $r := QuantitativeSimulation(M)$
    $T := T \cup r$

ChooseModel (F)
  for every interval parameter and
    qualitative variable p of F
    $v := ChooseValue(Domain(p))$

Substitute p by v in M
for every function h
H:=Choose H(h)
Subsitute h by H in M

where N is the number of simulations to be carried out. It is defined in accordance with section *Theoretical study of the conclusions*. Therefore, N is the number of trajectories in T.

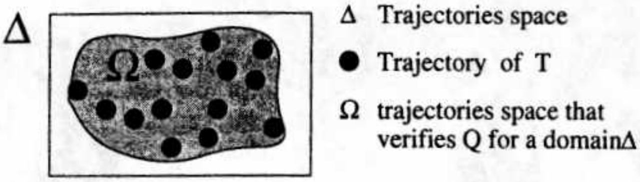

Figure 6: Trajectories space

Query/classification language

We propose a language to carry out queries and to classify with labels the database T. Therefore, this language allows us to classify the behaviour patterns of the system.

Abstract Syntax of the language

Let T be the set of all trajectories r stored in the database. The abstract syntax of the proposed language is

Q:	$\forall r \in T \bullet [r, P]$	P:	P_b
|	$\exists r \in T \bullet [r, P]$	|	$P \wedge P$
|	$\mathcal{N} r \in T \bullet [r, P]$	|	$P \vee P$
|	$[r, p]$	|	$\neg P$
P_b :	P_d	P_d :	EQ
|	$f(L(F))$	|	CL
|	$\forall t : F \bullet F$		
|	$\exists t : F \bullet F$		
F :	F_b	F_b :	e_b
|	$F \,\&\, F$	|	$e \in I$
|	$F \mid F$	|	$u(e)$
|	$!F$	|	$b(e, e)$

Table 2: Abstract syntax

A query Q on the database T may be a quantifier $\forall, \exists, \mathcal{N}$ applied on T, or a basic query $[r, P]$ that evaluates to *true* when the trajectory r verifies the property P. This property may be formulated by means of the composition of other properties using the boolean operators $\wedge, \vee, \neg$ and its result is the application of these operators among the partial properties.

A basic property P_b may be: a predefined property P_d, a boolean function f applied to a list L of points or intervals which verify the formula F, or a quantifier operator $\forall, \exists$ applied to the values of a particular trajectory for a time t. This time corresponding either to an instant of time, to a temporal range, or a set of temporal ranges related by a logical formula.

A defined property P_d is that one whose formulation is automatic. They are queries commonly used in dynamical systems. There are two predefined: EQ, which is verified when the trajectory ends up in a stable equilibrium; and CL that it is verified when it ends up in a limit cycle.

A formula F may be composed of other formulas combined by means of boolean operators &, |, ! and its result is the application of these operators among the partial formulas.

A basic formula F_b may be: a boolean expression e_b, or if a numeric expression e belongs to an interval I, or a unary u or binary b qualitative operator.

Classification

A classification rule is formulated as a set of basic queries with labels and possibly other expressions. A classification problem is proposed in accordance with the following abstract syntax:

$$[r, P_A] \Rightarrow A, e_{A1}, \ldots \quad [r, P_B] \Rightarrow B, e_{B1}, \ldots \quad \ldots$$

Semantics of the language

The semantics of every proposed statement is translated into a query on the database. A query $[r, P]$ is *true* when trajectory r verifies property P. Let be a query with a quantifier. Its semantics depends on its related quantifier. If it is $\forall$, a boolean value *true* is returned when all the trajectories $r \in T$ verify P. To prove that an existentially quantified statement is *true* is necessary to find at least one trajectory $r \in T$ that verifies the property P. If the quantifier is $\mathcal{N}$ then the number of trajectories of T that verify P is returned.

Let $\forall t : F_1 \bullet F_2$ be a basic property. This property is *true* if during the time that F_1 is satisfied, all the values of r verify F_2. On the other hand, if the quantifier is $\exists$ then it evaluates as *true* when at least a value of r that satisfies F_1, also satisfies F_2. In order to evaluate a formula F, it is necessary to substitute its variables for their values. These values are obtained from T.

Let $[r, P_A] \Rightarrow A, e_{A1}$ be a classification rule. A trajectory $r \in T$ is classified with the label A if it verifies property P_A. It is also included into the database for this trayectory the result of the evalutation of e_{A1}. This process is repeated with every classification rule.

Theoretical study of the conclusions

In this section, it is analysed if the obtained conclusions are applicable to the real system, that is, if the following affirmations would be concluded: "*all the behaviours of the system verify the property P*", or "*there is a behaviour of the system that verifies the property P*".

Firstly, it is necessary to answer the following question: *what is the necessary condition to ensure that all the behaviours of the system verify a property P?*.

Let $Vol(s)$ be the volume of a space s. Let Δ be the trajectory space of the system, an let Ω be the space of those trajectories of Δ that verify P (figure 6). Therefore, our goal is to solve the question *what is the condition that should be verified in order to guarantee that* $Vol(\Delta) = Vol(\Omega)$*?*. In a schematic way, we are interested in knowing the condition to guarantee that the following implication is true

$$\forall r \in T \bullet [r, P] \overset{\alpha}{\Rightarrow} \forall r \in \Delta \bullet [r, P] \qquad (10)$$

where α is the confidence degree. Statistical techniques are necessary to carry out this implication.

Let p be the probability that a trajectory r verifies a property P and $q = 1 - p$. Therefore p is

$$p = \frac{Vol(\Omega)}{Vol(\Delta)} \qquad (11)$$

Let a group of n trajectories be, and let x be a random variable whose value is n if the $n-1$ first trajectories verify P, and the $n-th$ does not. Let α be the confidence degree. The expression

$$\alpha = Probability(x > n) \qquad (12)$$

is the probability that the n first trajectories verify P and there is a trajectory that does not verify P among the rest of trajectories of Δ.

Therefore the probability p verifies

$$p \geq 1 - \frac{1}{n\alpha} \qquad (13)$$

Proof:
The expected value of a random variable x, denoted $E[x]$, is defined as follows

$$E[x] = \sum_{n=1}^{\infty} np^{n-1}q \qquad (14)$$

if we carry out symbolic manipulation in (14)

$$E[x] = \frac{q}{p}\sum_{n=1}^{\infty} np^{n} \qquad (15)$$

Replacing the geometric sum by its value, we obtain

$$E[x] = \frac{q}{p}\frac{p}{(1-p)^2} = \frac{1}{1-p} \qquad (16)$$

On the other hand, and if we apply the inequality of Chebyshev

$$E[x] = \sum_{x=1}^{\infty} xp(x) \geq \sum_{x=n+1}^{\infty} np^{n-1} = n \sum_{x=n+1}^{\infty} p^{n-1} \qquad (17)$$

Replacing the sum by its value, it is obtained

$$E[x] \geq n \; Probability(x > n) \qquad (18)$$

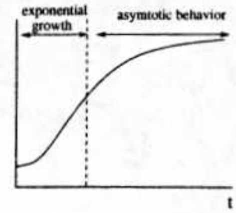

Figure 7: Logistic growth curve

Replacing $E[x]$ by its obtained value in (16), and manipulating the expression

$$\frac{1}{n(1-p)} \geq Probability(x > n) \qquad (19)$$

By (12)

$$\frac{1}{n(1-p)} \geq \alpha \qquad (20)$$

It follows that

$$\frac{1}{n\alpha} \geq 1 - p \Rightarrow p \geq 1 - \frac{1}{n\alpha} \qquad (21)$$

□

The obtained result (13) means that: *given a confidence degree* α*, if we want to ensure that a property P is true for a dynamic system with a probability p, it is necessary to obtain at least n trajectories verifying it.*

Next table shows several examples for the values of α, p, and n

$\alpha = 0.05$		$\alpha = 0.01$	
$p = 0.6$	$n = 50$	$p = 0.5$	$n = 200$
$p = 0.8$	$n = 100$	$p = 0.9$	$n = 1000$
$p = 0.98$	$n = 1000$	$p = 0.99$	$n = 10000$
$p = 0.998$	$n = 10000$	$p = 0.9999$	$n = 10^6$

Table 3: Examples for α,p and n

In the same way, and applying the generalized De Morgan laws for logic, an existentially quantified query $\exists r \in \Delta \bullet [r, P]$ may always be formulated as the universally quantified query $\neg\forall r \in \Delta \bullet [r, \neg Q]$. Therefore, the previous study may also be applied for the existential quantified queries.

Application to a logistic growth model with a delay

It is very common to find growth processes in which an initial phase of exponential growth is followed by another phase of approaching to a saturation value asymptotically (figure 7). These are given the following generic names: logistic, sigmoidal, and s-shaped processes. This growth is exhibited by systems for which exponential expansion is truncated by the limitation of the resources required for this growth. This behaviour is due to a positive feedback that is dominant in the

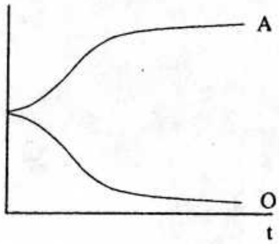

Figure 8: Logistic growth model

initial phase, and a negative feedback that is dominant in the final phase.

In the literature, these models have been profusely studied. They abound both in natural processes, and in social and socio-technical systems. They appear in the evolution of bacteria, in mineral extraction, in world population growth, and in economic development. Learning curves also show this type of behaviour.

The same thing happens with some diffusion phenomena within a given population, such as epidemics or rumors. Other examples of this behaviour are a population that grows in a habitat with limited resources, a technological innovation that is being introduced, or a new product that is being put on the market. In all these cases, their common behaviours are shown in figure 8. There is a bimodal behaviour pattern attractor: A stands for normal growth, and O for decay. It can be observed how it combines exponential with asymptotic growth. This phenomenon was fist modeled by the Belgian sociologist P. F. Verhulst in relation with human population growth. Nowadays, it has a wide variety of applications, and some of them have just been mentioned.

Let S be the qualitative model. If we add a delay in the feedback paths of S, then its differential equations are

$$\Phi \equiv \begin{cases} \dot{x} = x(n\,r - m), \\ y = delay_\tau(x), \quad x > 0, \quad r = h_1(y), \\ h_1 \equiv \{(-\infty, -\infty), +, (d_0, 0), +, (0, 1), \\ \quad +, (d_1, e_0), -, (1, 0), -(+\infty, -\infty)\} \end{cases}$$

being n the increasing factor, m the decreasing factor, and h_1 a qualitative function with a maximum point at (x_1, y_0) (see figure 9).

The initial conditions are

$$\Phi_0 \equiv \begin{cases} x_0 \in [LP_x, MP_x], \\ LP_x(m), \\ LP_x(n), \\ \tau \in [MP_\tau, VP_\tau] \end{cases}$$

where LP, MP, VP are qualitative unary operators for x, τ variables.

We would like to know:
1. if an equilibrium is always reached
2. if there is an equilibrium whose value is not zero

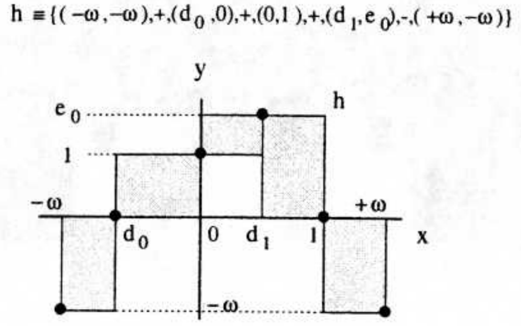

Figure 9: Qualitative function h_1

3. if all the trajectories with value zero at the equilibrium are reached without oscillations.
4. To classify the database in accordance with the behaviours of the system.

The methodology is applied to this model. Firstly, it is necessary to define the intervals associated with every qualitative operator:

$$\begin{array}{ll} LP_x = [0, 1] & MP_x = [1, 3] \\ MP_\tau = [0.5, 4] & VP_\tau = [4, 10] \end{array}$$

Next, the described transformation rules are applied to S, and it is obtained the following set of constraints

$$\begin{cases} \dot{x} = x(n\,r - m), \\ y = delay_\tau(x), \; x > 0, \; r = H_1(y) \\ H_1, \; x_0 \in [0, 3], \; m, n \in [0, 1], \; \tau \in [0.5, 10] \end{cases} \tag{22}$$

where the set H_1 has been obtained by applying *Choose H* to h_1. The algorithm *Database generation T* returns the trajectory database T.

Applying the proposed language, the proposed queries are formulated as follows:
1. $\forall r \in T \bullet [r, EQ]$
2. $\exists r \in T \bullet [r, EQ \wedge \exists t : t \simeq t_f \bullet ! AP0_x(x)]$
3. $\forall r \in T \bullet [r, EQ \wedge \exists t : t \simeq t_f \bullet AP0_x(x) \wedge$
$length(\dot{x} = 0) = 0$

The list of points where $\dot{x} = 0$ is the list with the maximum and minimum points. There are no oscillations when its length is 0.

The answers to the proposed questions were:
1. $True$, all trajectories of T reach a stable equilibrium. Therefore, we conclude *there is no cycle limit.*
2. $True$, some trajectories of T reach an equilibrium whose value is not zero. Therefore this is the first behaviour we have obtained. We know it as *recovered equilibrium.*
3. $False$, there are at lest two ways to reach this equilibrium: with oscillations (this behaviour is called as *retarded catastrophe*) and the other way is without oscillations (that it is called as *decay and extinction*).

In short, we have found the three possible behaviours patterns of the system (figure 10) and therefore, the proposed classification may be carried out as follows
$[r, EQ \wedge length(\dot{x} = 0) > 0 \wedge$

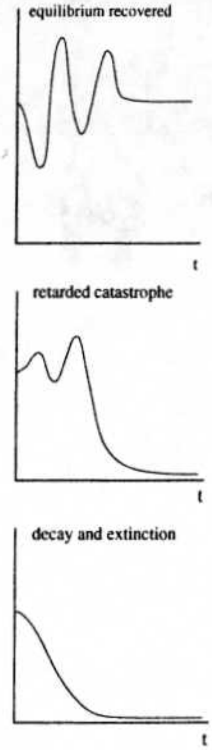

Figure 10: Logistic growth model with a delay

$$\exists t : t \simeq t_f \bullet !AP0_x(x)] \Rightarrow \mathit{recovered},$$
$$[r, EQ \wedge \mathit{length}(\dot{x} = 0) > 0 \wedge$$
$$\exists t : t \simeq t_f \bullet AP0_x(x)] \Rightarrow \mathit{retarded},$$
$$[r, EQ \wedge \exists t : t \simeq t_f \bullet AP0_x(x)] \Rightarrow \mathit{extinction},$$

The database T is labeled with three labels: *recovered, retarded* and *extinction*. All trajectories of T are labeled just by one label. Therefore, we may conclude that all patterns behaviours of the model have been classified. The results obtained with this methodology are in accordance with others appeared in the bibliography (Aracil et al. 1997) and (Karsky, Dore and Gueneau 1992) where the results are concluded by means of a mathematical reasoning. This cirscumstance encourages us to continue developing this methodology and to apply it to other systems with qualitative and quantitative knowledge.

Conclusions and further work

In this paper, a new methodology to automatize the analysis of dynamic systems with qualitative and quantitative knowledge has been presented. This methodology is based on a transformation process, application of stochastic techniques, quantitative simulation, and definition of a query/classification language.

The simulation results are stored into a quantitative database that may be queried and classified by means of the proposed language. Once the database is classified, genetic algorithms may be applied to obtain conclusions about the dynamic system.

There is enough bibliography that studies stationary states of dynamic systems. However, the study of transient states is also necessary either for natural and engineering systems.

In the future, the query/classification language must be enriched with operators for comparing trajectories, types of equations, etc. Dynamic systems with constraints and with multiple scales of time are also one of our future points of interest.

References

Aracil J.; Ponce E.; and Pizarro L. 1997. Behavior patterns of logistic models with a delay *Mathematics and computer in simulation* 44:123–141.

Berleant D., and Kuipers B.J. 1997 Qualitative and quantitative simulation: bridging the gap. *Artificial Intelligence* 95: 215–255

Bonarini A. and Bontempi G. 1994 Qua.SI.: Qualitative Simulation approach for fuzzy models. In *Proceedings of the 1994 Modelling and Simulation Multiconference,* 420–424, Barcelona, Spain.: Society for Computer Simulation International.

Bousson K. and Travè-Massuyés L. 1994 Putting more numbers in the qualitative simulator CA-EN. In *Proceedings of the Second International Conference on Inteligent Systems Engineering.* Hamburgo: International Conferences on Intelligent Systems Engineering, Inc.

Corliss G. F. 1995 Guaranteed error bounds for ordinary differential equations. *Theory of Numerics in Ordinary and Partial Differential Equations.* Oxford, UK.: Oxford University Press.

Gasca R.M. 1998 Razonamiento y Simulación en Sistemas que integran conocimiento cualitativo y cuantitativo. Ph.D. diss., Dept. of Computer Science, Seville Univ.

Hickey T.J. 1994 CLP(F) and Constrained ODEs. Technical Report. Michtom School of computer Science. Volen Center for Complex Systems. Brandeis Univ.

Kay H. 1996. Refining imprecise models and their behaviors. Ph.D. diss., Texas Univ.

Karsky M.; Dore J.-C.; and Gueneau P. 1992, Da la possibilité d'apparition de catastrophes différés. *Ecodecision No 6*

Kuipers B.J. 1994 *Qualitative reasoning. Modeling and simulation with incomplete knowledge,* Cambridge, Massachussetts.: The MIT Press.

Ortega J.A.; Gasca R.M.; and Toro M. 1998a. Including qualitative knowledge in semiqualitative dynamical systems. *Lecture Notes in Artificial Intelligence* 1415: 329–338.

Ortega J.A.; Gasca R.M.; and Toro M. 1998b. Obtaining semi-qualitative behaviors of dynamic systems by means of quantitative simulations. In *Workshop Model based systems and qualitative reasoning of the Thirtieth European Conference on Artificial Intelligence*, 21-26. Brigthon, UK: Supported by MONET, The European Network of Excellence in MBS & QR.

Aguilar J.; Riquelme J.; and Toro M. 1998. *Decision queue classifier for supervised learning using rotated hyperboxes. Lecture Notes in Artificial Intelligence* 1484: 326– 336.

Travé-Massuyés L.; Dague Ph.; and Guerrin F. 1997. *Le raisonement qualitativ pour les sciences de l'ingénieur.* Paris, France: Hermes Ed.

Vescovi M.; Farquhar A.; and Iwasaki Y. 1995. Numerical interval simulation: combined qualitative and quantitative simulation to bound behaviors of non-monotonic systems. In *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence*, 1806–1812, Montreal, Canada.: International Joint Conferences on Artificial Intelligence, Inc.