STDEVS. A NOVEL FORMALISM FOR MODELING AND SIMULATION OF STOCHASTIC DISCRETE EVENT SYSTEMS.

Ernesto Kofman and Rodrigo Castro

Laboratorio de Sistemas Dinámicos. FCEIA - Universidad Nacional de Rosario. Riobamba 245 bis - (2000) Rosario

Abstract: This article presents an extension of the DEVS formalism to represent stochastic discrete event systems. Making use of probability space theory, the resulting STDEVS formalism provides a formal framework for modeling an simulation of non–deterministic DEVS models.

Keywords: Discrete Event Systems, DEVS, Stochastic Systems, Probability Spaces.

1. INTRODUCTION

The DEVS formalism was developed by Zeigler in the mid–seventies Zeigler (1976); Zeigler et al. (2000).

Being a general system theoretic based formalism, DEVS allows one to represent all the systems whose input/output behavior can be described by sequences of events. Thus, discrete event systems modeled by Finite State Automatas, Petri Nets, Grafcets, Statecharts, etc., can be also represented by DEVS models (Zeigler and Vahie, 1993). Moreover, discrete time systems can be also represented by DEVS (Zeigler et al., 2000).

The generality of DEVS converted it into a widely used language to describe and to simulate many classes of discrete systems. A novel family of numerical integration algorithms was also developed allowing the simulation of continuous systems in term of DEVS (Kofman, 2004; Cellier and Kofman, 2006), exhibiting important advantages over discrete time approximations, mainly in the simulation of hybrid systems.

Consequently, many DEVS–based modeling and simulation software tools have been developed in recent years (Kim, 1994; Cho and Cho, 1997; Zeigler and Sarjoughian, 2000; Wainer et al., 2001; Filippi et al., 2002; Kofman et al., 2003).

The mentioned generality of DEVS comes from the fact that it permits modeling system with a set of infinite possible states, and where the new state after an event arrival may depend on the continuous elapsed time in the previous state. Models showing this complex behavior cannot be represented by any other discrete formalism.

However, being such a general purpose modeling tool has also its negative consequences. In fact, it is difficult to perform any kind of theoretical analysis on DEVS models. Also, modeling with DEVS is not a simple task. Anyway, automatic tools that translate finite state machines and Petri Nets into DEVS have been developed (Jacques and Wainer, 2002; Zheng and Wainer, 2003).

Another drawback of DEVS is that it is only formally defined for deterministic systems. Although some stochastic models where simulated with DEVS (Kofman and Junco, 2005), there is not a formal theoretic support for this kind of systems under the DEVS formalism.

Stochastic models play a fundamental role in discrete event system theory. In fact, any system involving uncertainties, unpredictable human actions and machine failures requires a non–deterministic treatment. Examples of stochastic discrete event formalisms are Markov Chains, Queuing Networks (Cassandras, 1993) and Stochastic Petri Nets (Ajmone Marsan et al., 1995). These tools permit analyizing and simulating stochastic models in several applications.

This paper is aimed to extend the DEVS formalism for modeling of stochastic systems. To this end, we introduce a new formalism called STDEVS. Taking into account that DEVS can work with sets of infinte possible states, we shall make use of Probability Space Theory to define the new formalism.

We also prove that STDEVS is a generalization of DEVS, i.e., DEVS is a particular case of STDEVS. This fact will allow us to combine DEVS and STDEVS models in coupled models of combined stochastic and deterministic systems.

Finally, we provide some simple examples to illustrate the usage of the new formalism.

The paper is organized as follows: in Section 2 we recall the basic concepts of the DEVS formalism and probability space theory. Then, in Section 3 we introduce the STDEVS formalism and we study its relation with the DEVS formalism. In Section 4 we present some simple examples and finally, in Section 5, we sketch some conclusions and discuss about future directions of this work.

2. PRELIMINARIES

2.1 DEVS Formalism

A DEVS model processes an input event trajectory and –according to that trajectory and to its own initial conditions– provokes an output event trajectory.

Formally, a DEVS *atomic* model is defined by the following structure:

$$M = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta),$$

where

- X is the set of input event values, i.e., the set of all the values that an input event can take;
- Y is the set of output event values;
- S is the set of state values;
- δ_{int}, δ_{ext}, λ and ta are functions which define the system dynamics.

Each possible state s ($s \in S$) has an associated *time advance* calculated by the *time advance function* $ta(s)$ ($ta(s) : S \to \mathbb{R}_0^+$). The *time advance* is a nonnegative real number saying how long the system remains in a given state in absence of input events.

Thus, if the state adopts the value s_1 at time t_1, after $ta(s_1)$ units of time (i.e., at time $ta(s_1)+t_1$) the system performs an *internal transition*, going to a new state s_2. The new state is calculated as $s_2 = \delta_{\text{int}}(s_1)$, where δ_{int} ($\delta_{\text{int}} : S \to S$) is called *internal transition function*.

When the state goes from s_1 to s_2 an output event is produced with value $y_1 = \lambda(s_1)$, where λ ($\lambda : S \to Y$) is called *output function*. Functions ta, δ_{int}, and λ define the autonomous behavior of a DEVS model.

When an input event arrives, the state changes instantaneously. The new state value depends not only on the input event value but also on the previous state value and the elapsed time since the last transition. If the system goes to the state s_3 at time t_3 and then an input event arrives at time t_3+e with value x_1, the new state is calculated as $s_4 = \delta_{\text{ext}}(s_3, e, x_1)$ (note that $ta(s_3) > e$). In this case, we say that the system performs an *external transition*. Function δ_{ext} ($\delta_{\text{ext}} : S \times \mathbb{R}_0^+ \times X \to S$) is called the *external transition function*. No output event is produced during an external transition.

DEVS models can be coupled. One of the most used coupling schemes for DEVS models includes the use of ports. Here, the external transition functions of the atomic models distinguish the value and arrival port of the events to calculate the next state. Similarly, the output functions produce output events which carry a value through a given port. Then the coupling basically consists in connections from output ports to input ports of different atomic models.

2.2 Probability Spaces

We recall here some concepts of probability spaces (Gray and Davisson, 2004).

A sample space S of a random experiment is a set that includes all the possible outcomes of the experiment.

An event space (also referred as *sigma–field* or *sigma–algebra*) $\mathcal{F}$ of the sample space S is a nonempty collection of subsets of S. The entries $F \in \mathcal{F}$ are called *events*. In order not to mix the concepts of event as a change on a system and this new definition of event, we shall refer to $\mathcal{F}$ as a sigma–field.

A sigma–field cannot be any arbitrary collection of subsets of S. A collection $\mathcal{F}$ must satisfy the following properties in order to constitute a sigma–field:

- if $F \in \mathcal{F}$ then $F^c \in \mathcal{F}$ (where F^c is the complement of F in S).
- if $F_i \in \mathcal{F}$ for $i = 1, \ldots, \infty$, then also

$$\bigcup_{i=1}^{\infty} F_i \in \mathcal{F}$$

Notice that since $F^c \cup F = S$, the last two conditions imply that $S \in \mathcal{F}$ and also $\phi \in \mathcal{F}$.

A particular sigma–field of S is the collection of all the subsets of S (this is called the power set of S). Another sigma–field of S is the one formed only by the sets S and ϕ.

Let $\mathcal{G}$ be a particular collection of subsets of the sample space S. The sigma–field generated by $\mathcal{G}$, denoted $\mathcal{M}(\mathcal{G})$, is the smallest sigma–field that contains all the elements of $\mathcal{G}$.

A pair $(S, \mathcal{F})$ consisting on a sample space S and a sigma field $\mathcal{F}$ of subsets of S is called a measurable space.

A probability measure P on a measurable space $(S, \mathcal{F})$ is an assignment of a real number $P(F)$ to every member F of the sigma-field, such that P obeys the following rules,

- Axiom 1. $P(F) \geq 0$ for all $F \in \mathcal{F}$.
- Axiom 2. $P(S) = 1$.
- Axiom 3. If $F_i \in \mathcal{F}$, $i = 1, \ldots, \infty$ are disjoint sets, then

$$P(\bigcup_{i=1}^{\infty} F_i) = \sum_{i=1}^{\infty} P(F_i)$$

When $\mathcal{F} = \mathcal{M}(\mathcal{G})$ (the sigma field is generated from a collection $\mathcal{G}$), the knlowledge of $P(G)$ with $G \in \mathcal{G}$ defines function P for every $F \in \mathcal{F}$.

Finally, a *probability space* is defined as a triple $(S, \mathcal{F}, P)$ consisting of a sample space S, a sigma–field $\mathcal{F}$ of subsets of S, and a probability measure P defined for all members of $\mathcal{F}$.

Synthesizing, for every $F \in \mathcal{F}$, $P(F)$ expresses the probability that the experiment produces a sample $s \in F \subseteq S$.

3. STDEVS FORMALISM

3.1 Definition of STDEVS

We define a STDEVS atomic model as a 8–*uple*

$$M = (X, Y, S, \mathcal{G}, P_{\text{int}}, P_{\text{ext}}, \lambda, ta)$$

where X, Y, S, λ and ta have the same definition than that of DEVS.

$\mathcal{G}$ is a collection of subsets of S. Defining, $\mathcal{F} = \mathcal{M}(\mathcal{G})$, the pair $(S, \mathcal{F})$ form a measurable space.

The *internal transition probability function* $P_{\text{int}} : S \times \mathcal{F} \to [0, 1]$ is defined so that, for every $s \in S$, function $P_{\text{int}}(s, \cdot)$ is a probability measure on the space $(S, \mathcal{F})$. Notice that since $\mathcal{F}$ is the sigma field generated by $\mathcal{G}$, it is enough to provide $P_{\text{int}}(s, G)$ for every $G \in \mathcal{G}$.

Similarly, the *external transition probability function* $P_{\text{ext}} : S \times \mathbb{R}_0^+ \times X \times \mathcal{F} \to [0, 1]$ is defined so that, for every $s \in S$, $e \in \mathbb{R}_0^+$ and $x \in X$, function $P_{\text{ext}}(s, e, x, \cdot)$ is a probability measure on the space $(S, \mathcal{F})$. As before, P_{ext} is completely determined defining $P_{\text{ext}}(s, e, x, G)$ for every $G \in \mathcal{G}$.

The behavior of the STDEVS model is similar to that of the DEVS model. The only difference is that the transition functions are not deterministic.

After an internal transition, the new state $\tilde{s}$ is not determined by an internal transition function. Instead, $P_{\text{int}}(s, G)$ gives the probability that the new state $\tilde{s}$ belongs to the set $G \in \mathcal{G}$, provided that the previous state was s.

Similarly, $P_{\text{ext}}(s, e, x, G)$ calculates the probability that the new state $\tilde{s}$ belongs to $G \in \mathcal{G}$ after receiving the input event x and provided that the elapsed time in the previous state s was e.

3.2 Coupling of STDEVS models

We define a coupled STDEVS model in the same way that coupled DEVS models are defined.

Conjecture 1. Closure under coupling

The coupling of n STDEVS models behaves like an equivalent atomic STDEVS model.

3.3 DEVS and STDEVS

Theorem 1. DEVS and STDEVS

Any DEVS model $M_D = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta)$ can be represented by an equivalent STDEVS model $M_S = (X, Y, S, \mathcal{G}, P_{\text{int}}, P_{\text{ext}}, \lambda, ta)$.

PROOF. Define the collection $\mathcal{G}$ as the power set of S (i.e., the set of all subsets of S). Thus, $\mathcal{F} = \mathcal{G}$.

Then, for every $F \in \mathcal{F}$, $s \in S$, $x \in X$ and $e \in \mathbb{R}_0^+$ we define

$$P_{\text{int}}(s, F) \triangleq \begin{cases} 1 & \text{if } \delta_{\text{int}}(s) \in F \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and

$$P_{\text{ext}}(s, e, x, F) \triangleq \begin{cases} 1 & \text{if } \delta_{\text{ext}}(s, e, x) \in F \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We shall prove first that $P_{\text{int}}(s, \cdot)$ and $P_{\text{ext}}(s, e, x, \cdot)$ are well defined, this is, they are probability measures on $(S, \mathcal{F})$. To prove this, we need to show that they satisfy Axioms 1,2 and 3.

Axiom 1 is satisfied since P_{int} and P_{ext} can only take the values 0 and 1.

Since $\delta_{\text{int}}(s) \in S$ and $\delta_{\text{ext}}(s,e,x) \in S$, it follows that $P_{\text{int}}(s,S) = 1$ and $P_{\text{ext}}(s,e,x,S) = 1$. Thus, Axiom 2 is also accomplished.

Let us take and arbitrary sequence of disjoint sets $F_i \in \mathcal{F}$. Notice that the element $\delta_{\text{int}}(s)$ can only belong to one of these sets (it is also possible that it does not belong to any of them).

If it belongs to one set, we have that

$$P(\bigcup_{i=1}^{\infty} F_i) = 1$$

since $\delta_{\text{int}}(s)$ belongs to the set union. On the other hand

$$\sum_{i=1}^{\infty} P(F_i) = P(F_j) = 1$$

where F_j is the only set containing $\delta_{\text{int}}(s)$. Thus, Axiom 3 is satisfied.

In case $\delta_{\text{int}}(s)$ does not belong to any of the sets, we have

$$P(\bigcup_{i=1}^{\infty} F_i) = 0 = \sum_{i=1}^{\infty} P(F_i)$$

Thus, Axiom 3 is also satisfied. Then, $P_{\text{int}}(s,\cdot)$ is a probability measure on $(S,\mathcal{F})$. Similarly, it results that $P_{\text{ext}}(s,e,x,\cdot)$ is a probability measure on the same space.

Notice that, being $\mathcal{F}$ the power set of S, the subset formed only by the element $\delta_{\text{int}}(s)$ belongs to $\mathcal{F}$. Thus, $P_{\text{int}}(s,\{\delta_{\text{int}}(s)\})$ is defined, and according to (1) it satisfies $P_{\text{int}}(s,\{\delta_{\text{int}}(s)\}) = 1$. Thus, when the STDEVS is in the state s it will evolve to $\delta_{\text{int}}(s)$ with probability 1.

Similarly, the subset $\{\delta_{\text{ext}}(s,e,x)\} \in \mathcal{F}$ and then $P_{\text{ext}}(s,e,x,\{\delta_{\text{ext}}(s,e,x)\})$ is defined, and according to (2) it is equal to 1. Consequently, when the STDEVS receives the event x afer staying in state s for e units of time, it will evolve to $\delta_{\text{ext}}(s,e,x)$ with probability 1.

Thus, the STDEVS model M_S behaves exactly as the DEVS model M_D. $\square$

This theorem proves that the DEVS formalism can be seen as a particular case of STDEVS.

This allows us to couple and combine STDEVS and DEVS models. For instance, when we have a coupling of deterministic and stochastic models, we can use DEVS to represent the former and STDEVS for the latter cases.

3.4 STDEVS Simulation

The only difference between DEVS and STDEVS is the way of calculating the new state after internal and external transitions. Thus, a STDEVS simulation engine is esentially a DEVS engine equipped with pseudo–random number generation capability.

PowerDEVS (Kofman et al., 2003) for instance, uses standard C++ libraries that contain a pseudo–random routine that generates numbers with uniform distribution. It is possible then to simulate a STDEVS model in PowerDEVS by using this routine to generate a pseudo–random numbers at the transition functions, and then to operate over them so that the resulting distributions coincide with those of P_{int} and P_{ext}.

4. EXAMPLES

We present here three examples of atomic STDEVS models. In the first one, the state (belonging to a continuous set)is randomly chosen after the internal transtion. In the second one, only one discrete component of the state is randomly chosen (after the external transition) and the internal transition is pure deterministic. In the third example, the state is a combination of discrete and continuous variables and it changes randomly.

4.1 A random memoryless workload generator

Consider a system that repeteadly generates a task. We shall assume that the time elapsed between consecutively generated tasks is random, with exponential distribution. If we call τ_k to the elapsed time between tasks k and $k+1$, it results that

$$P(\tau_k < t) = 1 - e^{-a\,t} \quad (3)$$

where $1/a$ is the mean time to the next generated task.

The following STDEVS model represents this behavior

$$M_G = (X, Y, S, \mathcal{G}, P_{\text{int}}, P_{\text{ext}}, \lambda, ta)$$

where

$$X = \phi,\ Y = \{(1\,out_1)\},\ S = \mathbb{R}_0^+$$
$$\lambda(s) = (1, out_1),\ ta(s) = s$$

Notice that we represent each task as an event carrying the value "1" by the output port out_1. Also, the state s is a nonegative number indicating the elapsed time to the next output event.

For each $t > 0$, we define the set $A_t \triangleq [0,t)$. We define also $\mathcal{G}$ as the collection of all the sets A_t.

The sigma–field results $\mathcal{F} = \mathcal{M}(\mathcal{G}) = \mathcal{B}(\mathbb{R})$ (i.e., the Borel space).

Then, the internal transtion probability function is

$$P_{\text{int}}(s, A_t) = 1 - e^{-a\,t}$$

In this way, the probability that the new state belongs to the interval $[0, t)$ is $1 - e^{-a\,t}$, according to (3).

Since the model does not receive events, we do not need to define function P_{ext}.

4.2 Workload Balancer

Consider a system that receives tasks and delivers them according to a selected stochastic process by its n output ports. Each task is represented by an event carrying a natural number in the set $T = \{1, 2, \ldots, m\}$ (there are m different tasks).

If we consider that the stochastic process is such that every task can be equiprobably delivered by any output port, then we can model this system as

$$M_B = (X, Y, S, \mathcal{G}, P_{\text{int}}, P_{\text{ext}}, \lambda, ta)$$

where

$$X = T \times \{inp_1\},\ Y = T \times \{out_1, \ldots, out_n\},$$
$$S = T \times \{1, \ldots, m\} \times \mathbb{R}_0^+$$
$$\lambda(w, p, \sigma) = (w, out_p),\ ta(w, p, \sigma) = \sigma$$

The state is a triplet $s = (w, p, \sigma)$, where w represents the last task received, p is the port where that task is delivered and σ is the time advance. After receiving an event with value x_v, the new state must be of the form $(x_v, \tilde{p}, 0)$. Notice that $\tilde{p}$ is the only randomly chosen state component.

We define $\mathcal{G} \triangleq T \times \{1, \ldots, m\} \times \{A_t\}$. Given an element of $\mathcal{G}$ of the form $G = (\tilde{w}, \tilde{p}, A_t) \in \mathcal{G}$, where $\tilde{w} \in T$ and $\tilde{p} \in \{1, \ldots, m\}$, function P_{ext} can be defined as

$$P_{\text{ext}}(s, e, x, G) \triangleq \begin{cases} 1/n & \text{if } (x_v, \tilde{p}, 0) \in G \\ 0 & \text{otherwise} \end{cases}$$

Function P_{int} is defined as

$$P_{\text{int}}(s, G) \triangleq \begin{cases} 1 & \text{if } (w, p, \infty) \in G \\ 0 & \text{otherwise} \end{cases}$$

notice that $P_{\text{int}}(s, G)$ is a fully deterministic transition function and it sets the time advance to ∞. In that way, no output event is provoked until a new event arrives.

4.3 Workload generator revisited

Let us modify the first example so that the system can generate m different tasks, where each task is represented by an event carrying a natural number in the set $T = \{1, 2, \ldots, m\}$.

For this example, we consider that each task can be equiprobably generated, and the choice of the task is independant on the elapsed time. The STDEVS model M_G can be then changed in the following way

$$X = \phi,\ Y = T \times \{out_1\},\ S = T \times \mathbb{R}_0^+$$
$$\lambda(w, \sigma) = (w, out_1),\ ta(w, \sigma) = s$$

Now the state is a pair $s = (w, \sigma)$, where w represents the task and σ the time advance. Both variables are randomly chosen after the internal transition.

The set collection is now $\mathcal{G} \triangleq T \times \{A_t\}$, and given an element $G = (\tilde{w}, A_t) \in \mathcal{G}$, with $\tilde{w} \in T$, we have

$$P_{\text{int}}(s, G) = \frac{(1 - e^{-a\,t})}{m}$$

In other words, $P_{\text{int}}(s, G)$ says that the probability that the new state has a duration $\tilde{\sigma}$ in the interval $[0, t]$ is $e^{-a\,t}$ and that the probability that the task chosen is $\tilde{w}$ is $1/m$ (the joint probability of both events is the product of them since they are independent processes.)

5. CONCLUSIONS

We presented STDEVS, a new formalism for modeling and simulation of stochastic discrete event system. STDEVS is an extension of DEVS that combines its system theoretic definition with probability space theory. Consequently, it provides a formal framework to treat general stochastic discrete event systems.

Future work should treat many open problems and check the usefulness of the formalism in complex applications.

Among the theoretical open problems, the conjecture of closure under coupling should be formally proven.

Another important issue is to provide conditions for legitimacy of STDEVS models. DEVS models are said to be illegitimate when they can perform an infinite number of transitions in a finite interval of time (Zeigler et al., 2000). Conditions on functions δ_{int} and ta exist so that it cannot happen. Finding similar conditions on P_{int} and ta in STDEVS constitutes another open problem.

Finally, it would be useful to develop a systematic way to calculate the new state according to P_{int} and P_{ext} based on the generation of pseudo–random uniform distributed numbers. Such a result would simplify the simulation of STDEVS on standard DEVS simulation engines.

Regarding applications, we are interested in exploring the modeling and simulation of complex Queuing Networks in interaction with continuous systems. Here, STDEVS can provide a unified framework since Continuous Systems can be approximated by DEVS (Cellier and Kofman, 2006).

Another application problem to be studied is that of the simulation of Stochastic Differential Equations, where STDEVS can be combined with quantization based integration methods (Kofman, 2004).

REFERENCES

M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1995.

Christos Cassandras. *Discrete Event Systems: Modeling and Performance Analysis*. Irwin and Aksen, Boston, Massachusetts, 1993.

F.E. Cellier and E. Kofman. *Continuous System Simulation*. Springer, New York, 2006.

Hyup Cho and Young Cho. *DEVS–C++ Reference Guide*. The University of Arizona, 1997.

J.B. Filippi, M. Delhom, and F. Bernardi. The JDEVS Environmental Modeling and Simulation Environment. In *Proceedings of IEMSS 2002*, volume 3, pages 283–288, 2002.

R. Gray and L. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, Cambridge, UK, 2004.

C. Jacques and G. Wainer. Using the CD++ DEVS toolkit to develop Petri Nets. In *Proceedings of the 2002 Summer Computer Simulation Conference*, San Diego, CA, 2002.

Tag Gon Kim. *DEVSim++ User's Manual. C++ Based Simulation with Hierarchical Modular DEVS Models*. Korea Advance Institute of Science and Technology, 1994.

E. Kofman. Discrete Event Simulation of Hybrid Systems. *SIAM Journal on Scientific Computing*, 25(5):1771–1797, 2004.

E. Kofman and S. Junco. DEVS as Part of and Integrating Tool in a Course on System Dynamics. In *Proceedings of ICSiE'05 (International Conference on Simulation in Education)*, 2005.

E. Kofman, M. Lapadula, and E. Pagliero. PowerDEVS:A DEVS Based Environment for Hybrid System Modeling and Simulation. Technical Report LSD0306, LSD, UNR, 2003. Available at www.fceia.unr.edu.ar/∼kofman.

G. Wainer, G. Christen, and A. Dobniewski. Defining DEVS Models with the CD++ Toolkit. In *Proceedings of ESS2001*, pages 633–637, 2001.

B. Zeigler. *Theory of Modeling and Simulation*. John Wiley & Sons, New York, 1976.

B. Zeigler, T.G. Kim, and H. Praehofer. *Theory of Modeling and Simulation. Second edition*. Academic Press, New York, 2000.

B. Zeigler and S. Vahie. Devs formalism and methodology: unity of conception/diversity of application. In *Proceedings of the 25th Winter Simulation Conference*, pages 573–579, Los Angeles, CA, 1993.

Bernard Zeigler and Hessam Sarjoughian. *Introduction to DEVS Modeling and Simulation with JAVA: A Simplified Approach to HLA-Compliant Distributed Simulations*. Arizona Center for Integrative Modeling and Simulation, 2000.

T. Zheng and G. Wainer. Implementing finite state machines using the CD++ toolkit. In *Proceedings of the 2003 Summer Computer Simulation Conference*, Montreal, QC. Canada, 2003.