Simulating Market Dynamics with CD++

Qi Liu, Gabriel Wainer

Department of Systems and Computer Engineering, Carleton University.
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6 Canada
{liuqi, gwainer}@sce.carleton.ca

Abstract. CD++ is an implementation of the Cell-DEVS formalism, which has been used to simulate various complex systems. In this study, we constructed a Cell-DEVS to simulate the dynamics of a dual market. Using new features of CD++, we obtained accurate results taking into account consumers' purchasing history. The resulting model allows fast execution, easier model implementation and maintenance.

1 Introduction

Cellular Automata (CA) [1] have become popular to simulate complex systems in a variety of research areas. CA are infinite n-dimensional lattices of cells updated synchronously according to local rules. Cell-DEVS [2], instead, uses the DEVS (Discrete Events Systems specifications) formalism [3] to define a cell space where each cell is defined as an atomic DEVS model. Each cell receives input external events from its neighboring cells and executes the events by evaluating local computing rules. The cell will change its states according to the execution results after a delay, and when it changes, it sends output messages to all its neighbors. CD++ [4] is an implementation of Cell-DEVS, which was recently extended to improve model definition, permitting to define more compact and flexible models [5]. We present a Cell-DEVS model to study the dynamics of markets [6]. The model simulates a dual market where consumers choose among competing products based on their preferences and the influence of others. A cell stands for a consumer who periodically renews the license with one of two Operating System providers. Three factors influence their behavior:

1. $U(c_{ij}, n, t)$, the **utility** consumer c_{ij} can obtain by using OS_n in time t,
2. $E(c_{ij}, n, t)$, the **network externality**, i.e., the influence of others; and
3. $P(c_{ij}, t)$, the **price** that consumer c_{ij} must pay.

The cell space is defined by a coupled Cell-DEVS, whose atomic cells are defined as C_{ij}=<X,Y,I,S,N,d,τ>.X={0,1,2} are the external inputs; Y={0,1,2} are the external outputs; S={0,1,2} are the states (0: non-user; i=1,2: user of OS_i). N={0,1,2} is the set of the inputs; d=100 is the transport delay for each cell. τ:N→S is the local computing function defined by Table 1 with $V(c_{ij},1,t)=U(c_{ij},n,t)+E(c_{ij},n,t)-P(c_{ij},t)$, for n=1 or 2 ($U$, E and P are computed as in [7], using the rules defined in [6]).

Table 1. Local computing rules

Result: 1	Rule : $V(c_{ij}, 1, t) > V(c_{ij}, 2, t)$ AND $V(c_{ij}, 1, t) > 0$
Result: 2	Rule : $V(c_{ij}, 2, t) > V(c_{ij}, 1, t)$ AND $V(c_{ij}, 2, t) > 0$
Result: 0	Rule : $0 > V(c_{ij}, 1, t)$ AND $0 > V(c_{ij}, 2, t)$

The model was implemented in CD++, and a set of experiments was carried out with six different settings. The tests were categorized into two groups: mature and new markets. The former group uses an initial cell space where the three possible states for each cell are uniformly distributed to represent a mature market; the latter group simulates a new market, in which only a few of cells represent new users. Fig. 1 shows the local computing rules for the new market and fluctuating price case.

```
%vo1 and vo2: skill accumulation for OS1 and OS2 respectively
%current state=0, vo1 & vo2 are depreciated by 0.25 at each time step
rule:{if((stateCount(1)+9*$vo1)>(stateCount(2)+9*$vo2),1,
       if((stateCount(2)+9*$vo2)>(stateCount(1)+9*$vo1), 2, 0))}
       {$vo1:=$vo1*0.25; $vo2:=$vo2*0.25;}  100  {(0,0)=0}
%current state=1: vo1 is incremented before depreciation
rule:{if((stateCount(2)+9*$vo2)>(stateCount(1)+9*$vo1),2,1)}
       {$vo1:=($vo1+1)*0.25; $vo2:=$vo2*0.25;}  100  {(0,0)=1}
%current state=2: vo2 is incremented by 1 before depreciation
rule:{if((stateCount(1)+9*$vo1)>(stateCount(2)+9*$vo2),1,2)}
       {$vo1:=$vo1*0.25; $vo2 := ($vo2+1)*0.25;}  100  {(0,0)=2}
```

Fig. 1. Definition of local computing rules in CD++: mature market and same price scenario

Three pricing strategies are defined: products with the same price, products with different prices, and products with fluctuating prices. The local computing rules are instantiated using the parameter values defined in Table 2.

Table 2. Parameter values for experimental frames

Settings	Mature Market			New Market		
	= Price	≠ Price	Fluctuating Price	= Price	≠ Price	Fluctuating Price
Parameters	U_{max}=.8 U_{min}=.4 θ=λ=.5 P_{OS1}=.25 P_{OS2}=.25	U_{max}=.8 U_{min}=.4 θ=λ=.5 P_{OS1}=.25 P_{OS2}=.3	U_{max}=.8,U_{min}=.4 θ=λ=.5 Q1=Q2=.1 $R1_{max}$=$R2_{max}$=.15 $R1_{min}$=$R2_{min}$=.05 μ1=.2, μ2=1	U_{max}=.8 U_{min}=.4 θ=λ=.5 P_{OS1}=.25 P_{OS2}=.25	U_{max}=.8 U_{min}=.4 θ =λ=.5 P_{OS1}=.25 P_{OS2}=.3	U_{max}=.8,U_{min}=.4 θ=λ=.5 Q1=Q2=0 $R1_{max}$=$R2_{max}$=.3 $R1_{min}$=$R2_{min}$=0 μ1=.2, μ2=1

2 Simulation Results

In this section we present simulation results for the different cases defined on Table 2. White cells represent non-users, light gray represent OS1, and dark gray OS2. The results in Fig. 2 show that non-users begin to use one of the two products with approximately equal probability, and users using the same products tend to aggregate together to form their own society, which in turn enhances network externality.

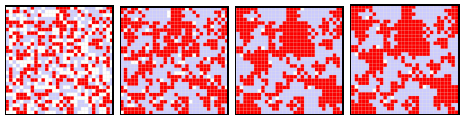

Fig. 2. Mature market and same price scenario

In Fig. 3, the price for OS1 is lower than the price for OS2 (all other parameters fixed), and most of the non-users choose to use OS1. Network externality again results in the aggregation of users.

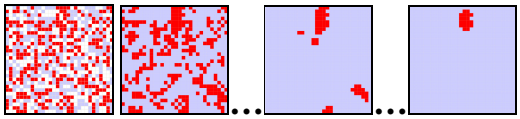

Fig. 3. Mature market and different price scenario

In Fig. 4, OS2 has higher pricing flexibility ($\mu 2 = 1$), while OS1 offers more rigid prices ($\mu 1 = 0.2$). As a result, OS2 gains bigger market share. If the local market shares for both products are equal, the price fluctuation disappears and network externality becomes the sole force in determining consumers' decisions.

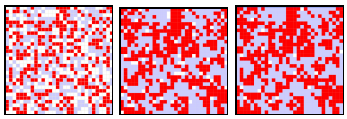

Fig. 4. Mature market and fluctuating price scenario

Fig. 5 shows that the development of the new market starts around the few initial users where the network externality takes effect. The number of users of both products rapidly grows almost at the same rate until the market is saturated. The initial users have been the pivots of a new market.

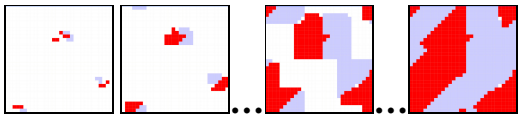

Fig. 5. New market and same price scenario

Fig. 6 shows how OS1 rapidly monopolizes the whole market by virtue of its lower prices (sensitivity of price is high in a new market). Fig. 7 shows that two types of new users ripple out from the initial ones into alternating circles. The development of the market exhibits a pattern that cannot be explained by any single factor of the value function.

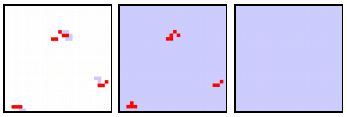

Fig. 6. New market and different price scenario

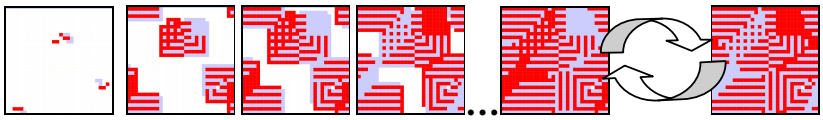

Fig. 7. New market and fluctuating price scenario

3 Conclusion

Cell–DEVS allows describing complex systems using an n-dimensional cell-based formalism. Timing behavior for the cells in the space can be defined using very simple constructions. The CD++ toolkit, based on the formalism, entitles the definition of complex cell-shaped models. We used CD++ to enhance a previous model for simulating market dynamics. The enhanced model obtained more accurate simulation results. Thanks to these new capabilities, we can achieve more efficient simulation, easier model implementation and maintenance. By running the simulation under different settings and analyzing the data generated, statistical results (long-run percentage of market shares, revenue and gross profit, parameters for pricing strategies etc.) can be obtained with sufficient precision. These results can guide us in predicting how the market will respond to various changes.

References

1. Wolfram, S. 2002. A new kind of science. Wolfram Media, Inc
2. G. Wainer, N. Giambiasi: N-dimensional Cell-DEVS. Discrete Events Systems: Theory and Applications, Kluwer, Vol.12. No.1 (January 2002) 135-157
3. B. Zeigler, T. Kim, H. Praehofer: Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press (2000).
4. G. Wainer: CD++: a toolkit to define discrete-event models. Software, Practice and Experience. Wiley. Vol. 32. No.3. (November 2002) 1261-1306
5. López, G. Wainer. Improved Cell-DEVS model definition in CD++. P.M.A. Sloot, B. Chopard, and A.G. Hoekstra (Eds.): ACRI 2004, LNCS 3305. Springer-Verlag. 2004.
6. S. Oda, K. Iyori, M. Ken, and K. Ueda, The Application of Cellular Automata to the Consumer's Theory: Simulating a Duopolistic Market. SEAL'98, LNCS 1585. pp. 454-461, Springer-Verlag. 1999.
7. Q. Liu, G. Wainer. Modeling a duopolistic market model using Cell-DEVS. Technical Report SCE-05-04. Systems and Computer Engineering. Carleton University. 2005.