PACIS 1997 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

December 1997

Calculating performance of business systems in information systems analysis

Ryo Sato
University of Tsukuba

Calculating performance of business systems in information systems analysis

Sato, Ryo
University of Tsukuba
rsato@shako.sk.tsukuba.ac.jp

Executive Summary

One of the competitive advantages in production systems is to have shorter lead time, which means the system reacts rapidly to customer's orders. The same principle holds true for most of business systems.

This short paper provides a way how to calculate the performance of business systems that is analyzed and designed through information systems analysis. The time how long it takes to accomplish a series of tasks is the measure of performance. Lead time is such an example. The calculation of the performance of business systems provides a qualitative measure to design agile business systems.

Business systems are often described through data flow diagrams and data models in information systems analysis. Since the DFD is convenient for drawing business systems, most information systems methodologies have adopted it. Sato and Praehofer (1997) has shown that a business system modeled by DFD's is a discrete event system called a business transaction system (BTS, for short). The DEVS formalism, originated by Zeigler (1976, 1984), is used to suitably describe the dynamic discrete-event behavior of business systems. If a DFD has the two characters below, then is called a BTS-DFD: First, there is a datastore between two processes. Second, each source is connected to a process via datastore. Many man-made systems can be modeled as BTS-DFD's. Examples are: sales and invoicing in retail companies, Just-in-time production in manufactures, medical diagnosis in hospitals, and servers in fast-food restaurants.

The max-plus algebra (Baccelli et al. 1992) is a linear algebra by which we can calculate the minimum time needed to finish a series of tasks of a BTS. According to Baccelli et al.(1992), we first show an example of the linear calculation in max-plus algebra.
And then, taking a Just-in-time assembly with Kanbans (i.e., data cards) as an example of business systems, we show how the minimum time required to get finished goods. Since discrete event mechanisms are common in man-made business systems, such a production system has similar characters with other discrete event mechanisms.

Below is the basic procedure of calculation.

(1) Make a DFD of your target business system. The DFD should be a BTS-DFD.
(2) Transform it into a timed event graph. And make the corresponding state transition equation in the max-plus algebra. (Or, directly make the equation from the DFD.)
(3) Solve the equation.

In this way, an example calculation of the lead time of a business system defined by a BTS-DFD is shown in the research, though the whole theory is not finished yet. Major deficit is that this calculation can not be easily applied to business systems with varying holding time, such as the processing of orders from customers. Even in such a case, if we use an average holding time for those order processing, the calculated lead time will give an performance index of a business system.

1. Introduction

This short paper provides a way how to calculate the performance of business systems that is analyzed and designed through information systems analysis. The time how long it takes to accomplish a series of tasks is the measure of performance. Lead time is such an example.

Business systems are described through data flow diagrams and data models. The former have takss (or, activities) which are interconnected with data flows and data stores. Since business systems modeled in that way are discrete event systems (Sato and Praehofer 1997, Sato 1997), the dynamics of such systems over time can be calculated by the max-plus algebra invented by Baccelli et al. (1992). That is, given initial inventories of intermediate materials, part, or request for ordered items between successive tasks, the performance of a business system is easily calculated by the max-plus algebra.

One of the competitive advantages in production systems is to have shorter lead time, which means the system reacts rapidly to customer's orders. In a manufacturing factory, once the orders in hand are scheduled, a new order will not be processed until the former inventory in the production line at that time finished. Therefore, the inventories in the production system should be decreased in order to get short lead time, while some amount of inventories are inevitable to make parallel operation of the processes possible. If every process in a production system consumes and produces intermediate products in the same rate, then inventories between two processes are not necessary. This principle is also true for many other business systems, though some inventories are not real things but those of services requested. In this sense, the calculation of the performance of business systems provides a qualitative measure to design agile business systems.

2. Discrete Event Model : BTS-DFD

Among the many existing tools for information systems analysis, the dataflow diagram (DFD) is one of the most widely-used diagrammatic forms for describing business systems. A DFD is a simple graphical network of dataflows, business processes, datastores and external organizations. The dataflow is a labeled arrow, the process a round rectangle, the datastore an open rectangle, and the sink or source a rectangle. While a DFD depicts the static appearance of a business system in terms of what constitutes that system, there are other tools for describing the system's dynamic aspect. The entity life history (ELH), which also has a diagrammatic form, is one such tool.

Since the DFD is convenient for drawing business systems, most information systems methodologies have adopted it. Sato and Praehofer (1997) has shown that a business system modeled by DFD's is a discrete event system called a business transaction system (BTS, for short). The DEVS formalism, originated by Zeigler (1976, 1984), is used to suitably describe the dynamic discrete-event behavior of business systems. If a DFD has the two characters below, then is called a BTS-DFD.

BTS-DFD : First, there is a datastore between two processes. Second, each source is connected to a process via datastore.

Many man-made systems can be modeled as BTS-DFD's. Examples are: sales and invoicing in retail companies, JIT production in manufactures, medical diagnosis in hospitals, and servers in fastfood restaurants.

3. Max-plus Algebra

Discrete event systems has two characters: Parallel processing and synchronization. The latter means dependency among a series of tasks that should be filled in a business. For example, an order for a commodity to a retailer should be followed pricing, enquiry of the inventory, delivery and then invoicing. Or, an production-order of finished goods to a manufacturer may start the assignment of raw materials that will be followed by a series of processing and assembly. A task for an order in a such series of tasks can not be started until the former task(or, tasks) finishes. This relationship is called synchronization.

The max-plus algebra (Baccelli et al. 1992) is a linear algebra by which we can calculate the minimum time needed to finish a series of tasks.According to Baccelli et al.(1992), we show an example of the linear calculation in max-plus algebra.

Assume we have a 2x2 matrix below. Then the multiplication to a vector from the left is calculated as follows.

$$\begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3\cdot 1 + 7\cdot 0 \\ 2\cdot 1 + 4\cdot 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 3\cdot 3 + 7\cdot 2 \\ 2\cdot 3 + 4\cdot 2 \end{pmatrix} = \begin{pmatrix} 9 + 14 \\ 6 + 8 \end{pmatrix} = \begin{pmatrix} 23 \\ 14 \end{pmatrix}$$

The max-plus algebra is a linear algebra, whose multiplication is usual plus "+" operation and addition is "to take the maximum" operation in real numbers. So we have

$$\begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \max(3+1, 7+0) \\ \max(2+1, 4+0) \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} \max(3+7, 7+4) \\ \max(2+7, 4+4) \end{pmatrix} = \begin{pmatrix} 11 \\ 9 \end{pmatrix}$$

If we denote the multiplication of max-plus algebra by $\otimes$, and the addition by $\oplus$, then we have the following "linear calculation."

$$\begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \otimes 1 \oplus 7 \otimes 0 \\ 2 \otimes 1 \oplus 4 \otimes 0 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix}$$

Here, addition will be calculated after possible multiplication, just as the usual linear algebra of real numbers. Thus, the meaning of the equation is different, and the appearance is the same with usual matrix calculation.

Since the multiplication $\otimes$ is the addition of reals, the unit element for $\otimes$ is 0, that is denoted by e. That is, for any x, $x \otimes e = e \otimes x = x$. Similarly, the unit for $\oplus$ is denoted by e, and $e = -\infty$. For any x, $x \oplus e = e \oplus x = x$ hold.

Mathematically speaking, the max-plus algebra has shown not to be a field (Baccelli et al. 1992). That is,

(1) < $\otimes$, e> is a commutative group;

(2) < $\oplus$, e >is a commutative monoid;

(3) $\otimes$ is distributive to $\oplus$: $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.

4. Example calculation for a business system

Taking a Just-in-time assembly with Kanbans (i.e., data cards) as an example of business systems, we show how the minimum time required to get finished goods. This is possible for the business systems each of which task takes the same amount of time for every job to be processed. (Different tasks in a business systems may have different constant amounts of time to be taken, as in the below example.)

Since discrete event mechanisms are common in man-made business systems, such a production system has similar characters with other discrete event mechanisms.

Below is the basic procedure of calculation.

(1) Make a DFD of your target business system. The DFD should be a BTS-DFD.
(2) Transform it into a timed event graph. And make the corresponding state transition equation in the max-plus algebra. (Or, directly make the equation from the DFD.)
(3) Solve the equation.

Now we proceed to a Just-in-time example. Fig. 1 is the BTS-DFD of the basic flow of materials in an assembly line.

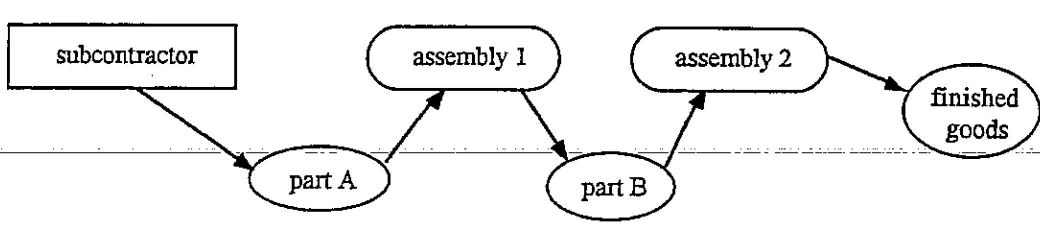

Fig. 1. Material flow in an assembly line

The assembly starts from the material purchasing and ends in the warehouse for finished goods.

Fig. 1 does not have any control. Fig. 2 is a Just-in-time control system with kanbans. Kanbans are cards and used between two sequential processes. One kanban corresponds to the small quantity of specific part or products. When prescribed number of kanbans are accumulated in its holding place for a kanban then an production-order is issued to its preceding process. In this way, orders for production are issued only to the final process in the production systems. The demand for the final product is the order to the final process and is the only input to the whole system. The kanban system is also called as Toyota system (Monden 1983).

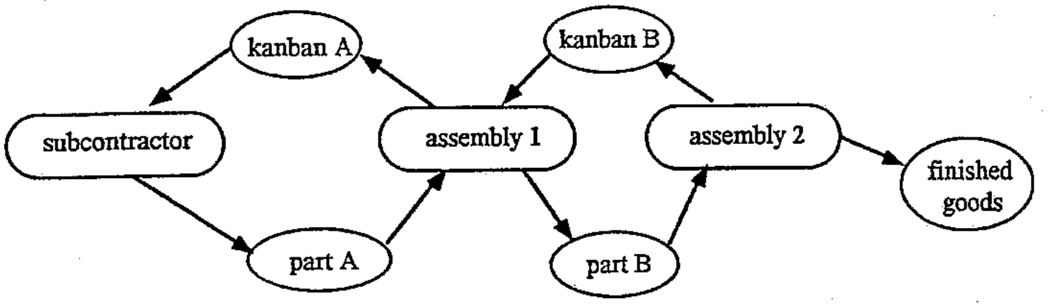

Fig. 2. Production control with kanbans.

The transformation from a BTS-DFD to a Petri net graph is easy. A data store shown by an oval in DFD will be a place, and a task shown by a round rectangular will be a combination of transition-place-transition. In the conbination, firing of the first transition means the starting of the taks's processing, while the last transition does the finish of it. The intermediate place represents the task's "busy state." (Fig. 3)

Each transition in the Petri net is numbered from the beginning, which indicates the preference to fire in the case that many transitions can fire at a time. For example, if t2 and t_5 are possible to fire at a time, then t_2 will fire first and then t_5 will do. Each

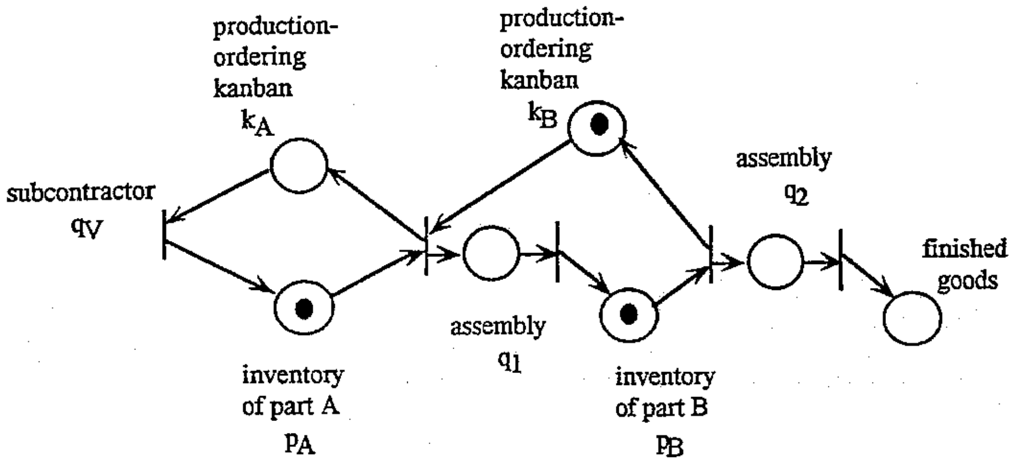

Fig. 3. Petri net graph generated from Fig. 2.

transition fires when there exist enough tokens (that represents, say, inventories of jobs) in the incoming places. Each activity needs some length of time to finish its processing after it started, and the needed time is modeled as the holding time of the place. The holding time of a place is assumed to be constant, though different places can have respective holding times. The holding times for places are affixed in Fig. 4.

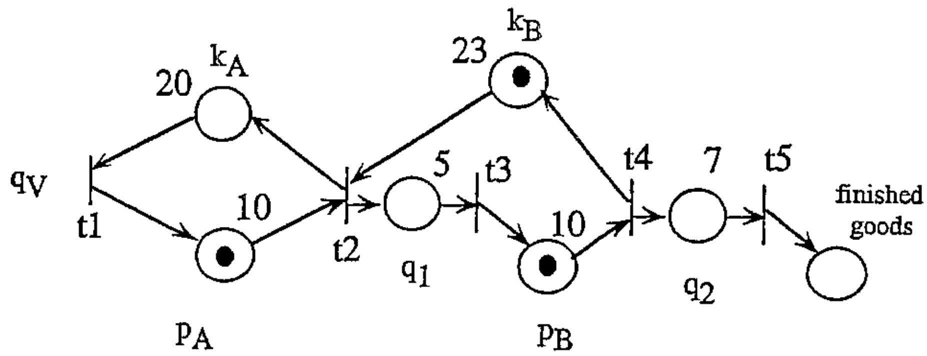

Fig. 4. A Petri net with tokens and holding times

Now, let X_i (k) denote the k -th epoch of a transition i for firing. The situation shown by Fig. 4 is represented by the set of following equations on the max-plus algebra, where usual notations for the multiplication and addition are employed.

$$
\begin{aligned}
X_1(t+1) &= 20\, X_2(t+1) \\
X_2(t+1) &= 23\, X_4(t) + 10\, X_1(t) \\
X_3(t+1) &= 5\, X_2(t+1) \\
X_4(t+1) &= 10\, X_3(t) \\
X_5(t+1) &= 7\, X_4(t+1)
\end{aligned}
$$

How to get equations

$X_1(t+1) = 20\, X_2(t+1)$

The transition t1 will fire for (t+1)-st time at the epoch when the (t+1)-st token entered to the place kA through t_2 and had been held there for 20 units of time. Please note that t and t+1 do not mean the time but counters to show how many times a transition fired.

$X_2(t+1) = 23\, X_4(t) + 10\, X_1(t)$

The epoch when t_2 fires for (t+1)-st time is the latest of the two epochs when the k-th token in p_A consumes its holding time after $X_1(t)$, and when the k-th token in k_B consumes its holding time after $X_4(t)$.

How to solve the simultaneous equation

If the set of equations are represented in a matrix form, and if the inverse of the "state transition" matrix can be obtained, then derivation of the solution will be straight forward. Fortunately, we do not have to get the inverse of a matrix as seen below.

$$
X(t+1) = \begin{pmatrix} \varepsilon & 20 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon \end{pmatrix} X(t+1) + \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & \varepsilon & 23 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} X(t)
$$

where

$$
X(t+1) = \begin{pmatrix} X_1(t+1) \\ X_2(t+1) \\ X_3(t+1) \\ X_4(t+1) \\ X_5(t+1) \end{pmatrix}
$$

Basically, we just substitute the righthand side of the above matrix equation for X(t+1) again and again. For simplicity, denote the equation above as follows.

$X(t+1) = A_0\, X(t+1) + A_1\, X(t)$

This gives the following.

$$X(t+1) = A_0 (A_0 X(t+1) + A_1 X(t)) + A_1 X(t)$$
$$= A_0^2 X(t+1) + (A_0A_1 + A_1) X(t)$$
$$= A_0^2 X(t+1) + (A_0 + E)A_1 X(t),$$

where E is the unit matrix whose diagonal elements are the unit e and others ε. Since

$$A_0^2 = \begin{pmatrix} \varepsilon & 20 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon \end{pmatrix} \begin{pmatrix} \varepsilon & 20 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon \end{pmatrix} = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

holds, $A_0^n = 0$ for each n, n > 1, where 0 is the null matrix all elements of which are zero ε.

$$A_0^n = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

Now, we have

$$(A_0)^* A_1 = (A_0 + E)A_1 = \begin{pmatrix} e & 20 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 7 & e \end{pmatrix} \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & \varepsilon & 23 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

$$= \begin{pmatrix} 10+20 & \varepsilon & \varepsilon & 23+20 & \varepsilon \\ 10 & \varepsilon & \varepsilon & 23 & \varepsilon \\ 10+5 & \varepsilon & \varepsilon & 23+5 & \varepsilon \\ \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 10+7 & \varepsilon & \varepsilon \end{pmatrix}$$

Therefore, the solution is the following.

$$X(t+1) = (A_0)^* A_1 X(t)$$

Then, for example, the next firing epoch for t_5 is given by multiplying the following C from the left.

$$C = (e, e, e, e, e)$$
$$C (A_0)^* A_1 = \begin{pmatrix} \varepsilon & \varepsilon & 10+7 & \varepsilon & \varepsilon \end{pmatrix}$$

The value of 10 + 7 = 17 means that the sum of holding time of part B and the required time for the second assembly activity in the original DFD.

5. Conclusion

An example calculation of the lead time of a business system defined by a BTS-DFD has been shown, though the theory is not finished yet.

Major deficit is that this calculation can not be easily applied to business systems with varying holding time, such as the processing of orders from customers. Even in such a case, if we use an average holding time for those order processing, the calculated lead time will give an performance index of a business system.

References

Zeigler, B. P. *Theory of modelling and simulation*. John Wiley, 1976.
Zeigler, B. P. *Multifaceted modeling and discrete event simulation*. Academic, 1984.
Baccelli, F.L., Cohen, G., Olsder, G.J., and Quadrat, J.P. *Synchronization and Linearity — An algebra for discrete event systems*. John Wiley, 1992.
Sato, R. and Praehofer, H. "A discrete event model of business system - A Systems Theoretic for Information Systems Analysis: Part 1." *IEEE Trans. Systems, Man, and Cybernetics*, Volume 27, January 1997, pp.1-10.
Sato, R. "Meaning of Dataflow Diagram and Entity Life History - A Systems Theoretic Foundation for Information Systems Analysis: Part 2." *IEEE Trans. Systems, Man, and Cybernetics*, Volume 27, January 1997, pp.11-22.
Monden, Y. *Toyota Production System - Practical approach to production management*. Industrial Engineering and Management Press, Institute of Industrial Engineers, 1983.