A (MAX,+) APPROACH FOR TIME IN MESSAGE SEQUENCE CHARTS

Pierre Le Maigat

Loïc Hélouët

IRISA, Campus de Beaulieu

35042 RENNES cedex, France

plemaiga@irisa.fr, helouet@irisa.fr

Abstract This paper details an approach for studying time in Message Sequence Charts (MSCs). MSCs are first transformed into order automata, and then into (max,+) automata, which allows for the use of well known (max,+) techniques.

Keywords: HMSC, (max, +)-automata, order automata, performance evaluation, timed systems.

1. INTRODUCTION

The use of scenarios allows for fast definition of applications, while staying at a very abstract level. Even if not detailed, they can be used for capturing architectural and behavioral requirements, and already allow analysis of the designed system. For these reasons, they have gained an increasing importance in methodologies such as UML (Unified Modeling Language).

Message Sequence Charts (MSCs) is a graphical formalism based on scenarios for distributed systems specification. Semantics of MSCs has been well studied, but the analysis of time within scenarios remains in its early days [1, 5, 7, 2]. However, considering performance analysis on high level specifications could allow refinements, and avoid costly modifications during further development stages. Some questions immediately arise: What happens when a process is delayed? From a given state of the system, what is the minimal delay before returning into this state? What is the duration of a given sequence? How long does each process takes to perform his task in a given sequence? What is the asymptotic behavior of a network ? Is the size of communication buffers bounded, and if not, what is the growth rate of each buffer?

This paper proposes a (max,+) modeling of time in Message Sequence Charts based on order automata [4], and their interpretation into timed diagrams (a particular (max,+) automaton), which allows to answer most of the above questions. This paper is organized as follows: Section 2 describes briefly Message Sequence Charts and how to introduce time within this formalism. Section 4 describes the translation of Message Sequence Charts into timed diagrams, and section 5 shows that most of the above questions can be solved using a (max,+) approach, before conclusion.

2. MESSAGE SEQUENCE CHARTS

A *Basic Message Sequence Chart* (bMSC) graphically defines the behavior of processes, called *instances*. An instance is represented by a vertical axis, along which events are ordered. A message exchange is represented by an arrow from the emitting instance to the receiving instance. Events can be sending events, receiving events, atomic actions,... Clearly, a bMSC defines a precedence relation: a message emission precedes its reception, and events are in a top-down order on the instance axis. So, a bMSC can be described by a labeled partial order $M =< E, \leq, \alpha, I, A >$, where: E is a set of events, $\leq$ is an order relation(transitive, antisymmetric, reflexive) on E, A is a set of action names, I is a set of instance names, and α is a labeling function: $\alpha : E \longrightarrow A \times I$. We will also denote by $\phi(e)$ the instance that performs event $e \in E$.

In order to study a timed behavior of Message Sequence Charts, a duration must be associated to each event and each communication. A timed bMSC can be described by a tuple $M_t =< E, \leq, A, I, \alpha, \Delta, T >$, where $E, \leq, A, I, \alpha$ have the same meaning as previously, and the map $\Delta : E \longrightarrow \mathbb{R}_+$ associates a duration δ_i to each event e_i, $T : E \times E \longrightarrow \mathbb{R}_+$ associates a duration τ_j to each message m_j: $T(e, e') = \tau_j$ iff $e :=!m_j$ is the emission of a message m_j, and $e' :=?m_j$ is the corresponding message reception. Consider bMSC M_2 of Figure 1. The durations associated to this diagram are: δ_3 (time for writing m_2 in a buffer) τ_2 (transiting time between A and B), δ_4 (time for extracting m_2 from a buffer), δ_5 (time for executing action a. The time needed for sending message m_2 from A to C is the time between the beginning of e_3 and then end of e_4: $\delta_3 + \tau + \delta_4$. We consider that an event is executed as soon as possible, i.e. when all its predecessors in the causal order are terminated.

bMSCs can be composed using a higher level formalism, called *High-level Message Sequence Charts*(HMSCs). This notation comports sequence, loop, alternative, and parallel composition operators, that are defined in [9]. Within this paper, we will only deal with a subset of HMSCs, including sequence, alternative, and loops. So, a HMSC can be seen as a kind of "bMSC automaton", and defined formally as a graph $H_t =< N, \longrightarrow, l, \mathcal{M}_t >$, where: N is a set of nodes, $\longrightarrow$ is a set of edges, l is a node labeling function, and $\mathcal{M}_t$ is a set of timed bMSCs.

For any timed bMSC $M \in \mathcal{M}_t$ of H, we will note $I(M)$ the set of instances in M. An example HMSC is given in Figure 1. This example comports a loop (sequence $M1$;$M2$ can be repeated infinitely), and an alternative(choice between the sequence $M1; M2$ and $M3$).

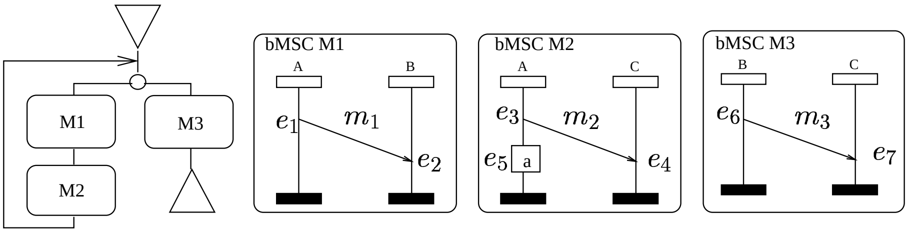


Figure 1 An example HMSC

A sequence of bMSCs do not impose any synchronization on events. Consequently, the meaning of a sequence, is just the partial order concatenation along the instance axis[1], and an event e can only be executed if all its predecessors have been executed. Time progression follows the same rule: on the example of Figure 1, the date of the first occurrence of e_5 is $t = \delta_1 + \delta_3$. Furthermore, HMSC in Figure 2 describes two different possible executions, depending on the durations δ_1 and δ_2. When $\delta_1 < \delta_2$, the time needed for a communication and the number of transiting messages increase, and when $\delta_1 > \delta_2$, they remain constant.

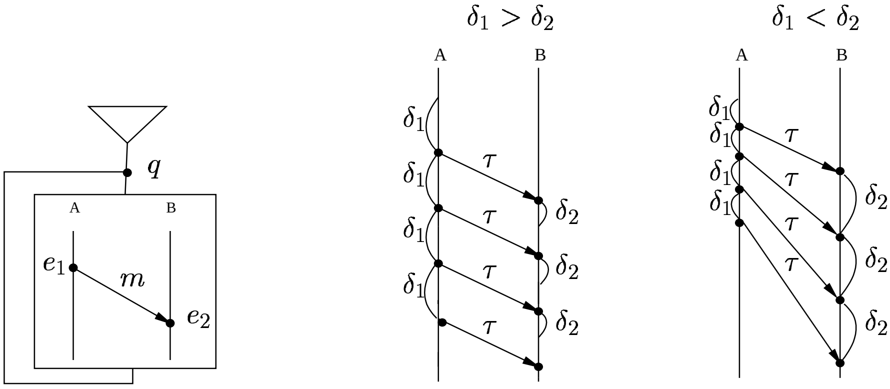


Figure 2 A timed HMSC and two possible behaviors

To analyze the asymptotic behavior of a HMSC, we propose the notion of asymptotic mean traffic, defined for a particular execution $U.V^\omega$ $(U, V \in \mathcal{M}_t^*)$ of a HMSC H by:

$$t_m = \frac{\text{number of message send by } V}{\text{average execution time (in stationary rate) of } V}$$

Modeling time within MSCs using order automata is a natural solution, which is described in next sections. Among other things, it allows to calculate t_m.

[1]Sequence can be seen as Pratt's local concatenation [6], where ϕ defines locality

3. ORDER AUTOMATA AND TIMED DIAGRAM

The set $(\mathbb{R} \cup \{-\infty\}, max, +)$ is an idempotent semiring, called $\mathbb{R}_{max}$. We will sometimes use the notations $\oplus$ and $\otimes$ to design the operations max and $+$. Let$\mathcal{M}_{n\times m}(\mathbb{R}_{max})$ be the semiring of matrices with coefficients in $\mathbb{R}_{max}$. A $(\max, +)$-automata is a tuple $\mathcal{A} = (\Sigma, Q, \alpha, T, \beta)$ where Σ is a finite set of letters, Q a finite set of states, T a map between $Q \times \Sigma \times Q$ and $\mathbb{R}_{max}$ and α,β two maps between Q and $\mathbb{R}_{max}$. Equivalentely, a $(\max, +)$-automata can be defined by its linear representation, (α, μ, β), with $\alpha \in \mathcal{M}_{Q\times 1}(\mathbb{R}_{max})$, $\beta \in \mathcal{M}_{1\times Q}(\mathbb{R}_{max})$ and $\mu : \Sigma^* \longrightarrow \mathcal{M}_{Q\times Q}(\mathbb{R}_{max})$ a morphism of monoids. Following [3], we call *dater* over Σ, a map $y : \Sigma^* \longrightarrow \mathbb{R} \cup \{-\infty\}$, that assigns a date to each execution $w \in \Sigma^*$. We can define, for a $(\max, +)$-automata, two types of daters: the internal dater, defined for all $q \in Q$, by $x_q = \alpha.\mu.\ {}^t(-\infty, \ldots, 0, \ldots, -\infty)$, with 0 in the q^{th} position; and the final dater $y = \alpha.\mu.\beta$.

Timed diagrams: In a HMSC, instances are considered as flows of events, and messages as causal relations between these flows. HMSCs describe compositions of these flows and constraints, that can be easily modeled by order automata [4]. These order automata are then translated into $(\max, +)$-automata (called timed diagrams) in order to analyze the temporal behavior of MSCs.

Definition 1 *An (pondered) order automata is given by a (classical) non-necessarily deterministic automata on Σ and a triplet $(\eta, \mathcal{O}, \sigma)$, where:*

- *Σ is a finite alphabet of actions and for all $a \in \Sigma$, $\mathcal{F}(a)$ is a finite set called alphabet of flows of a,*
- *$\mathcal{O}$ is a map assigning to each action $a \in \Sigma$ a bipartite order $(\mathcal{F}(a)_0 \cup \mathcal{F}(a)_1, \leq)$, and a ponderation δ such that:*

  - $\mathcal{F}(a)_0$ and $\mathcal{F}(a)_1$ are two disjoint copies of $\mathcal{F}(a)$ with $X_0 \leq X_1$ for all $X \in \mathcal{F}(a)$,

  - $\delta : \mathcal{F}(a)_0 \times \mathcal{F}(a)_1 \to \mathbb{R} \cup \{-\infty\}$ is a map, non-negative on the comparability graph of the order, such that for all X in the order $\delta(X, X) = 0$, and taking value $-\infty$ elsewhere,
- *$\eta : \bigcup_{a\in\Sigma} \mathcal{F}(a) \longrightarrow \mathbb{R}\cup\{-\infty\}$ (resp. $\sigma : \bigcup_{a\in\Sigma} \mathcal{F}(a) \longrightarrow \mathbb{R}\cup\{-\infty\}$) is a map called initial condition (resp. final condition).*

Figure 3 represents an order automata with alphabet of action $\Sigma = \{a, b, c\}$ and alphabets of flows $\mathcal{F}(a) = \{A, B\}$, $\mathcal{F}(b) = \{A, C\}$ and $\mathcal{F}(c) = \{B, C\}$ (the initial and final conditions are omitted).

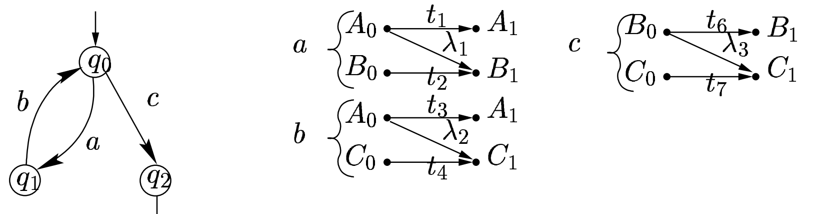


Figure 3 Order automaton

For $a \in \Sigma$, the elements of $\mathcal{F}(a)$ are flows, i.e sequences of values over a domain. A pondered bipartite order on these flows represents the temporal dependencies between two occurrences (consecutive or not) of the flows. As time must increase, we force $\delta(X_0, X_1) \geq 0$ for $X \in \mathcal{F}(a)$. For example, if d_A^n (resp. d_B^n) designs the date of the n^{th} occurrence of A (resp. of B), the order $\mathcal{O}(a)$ tells us $d_B^{n+k} \geq max(d_B^n + t_2, d_A^n + \lambda_1)$, with $k \geq 1$. Actions $a, b, c \cdots \in \Sigma$ define time constraints on flows, which can be extended to words.

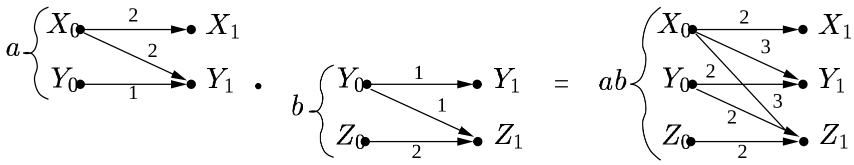
Definition 2 *Let* $\mathcal{O}(a) = (P, \delta_a)$ *and* $\mathcal{O}(b) = (Q, \delta_b)$ *be the pondered bipartite order associated to* a *and* b *in* Σ. *The sequence* $a.b$ *introduces new relations on the flows* $\mathcal{F}(ab) = \mathcal{F}(a) \cup \mathcal{F}(b)$. *The product* $\mathcal{O}(ab) =_{def} \mathcal{O}(a).\mathcal{O}(b)$ *between* $\mathcal{O}(a)$ *and* $\mathcal{O}(b)$ *is* $(\mathcal{F}(ab), \leq_P . \leq_Q, \delta_{ab})$ *where* $\leq_P . \leq_Q$ *is defined by:*

$$\leq_P . \leq_Q = \leq_P \cup \leq_Q \cup \{(x,y) \in \mathcal{F}(ab)^2, \exists z \in \mathcal{F}(a) \cap \mathcal{F}(b) \text{ such that } x \leq_P z \wedge z \leq_Q y\}$$

and, with the convention that a maximum over the empty set is equal to $-\infty$, *the map* δ_{ab} *is defined by:*

$$\delta_{ab}(X_0, Y_1) = \max\{\max_{Z \in \mathcal{F}_a \cap \mathcal{F}_b} \{\delta_a(X_0, Z_1) + \delta_b(Z_0, Y_1)\}, \delta_a(X_0, Y_1), \delta_b(X_0, Y_1)\}$$

We can define inductively the temporal relations between flows for any word $w = v.a \in \Sigma^*$ by $O(w) = O(v) \cdot O(a)$.



Definition 3 *Let* w *be a word of* Σ^* *and* $\mathcal{O}(w)$ *be its pondered bipartite order. For all flows* $Y \in \mathcal{F}(w)$, *we call* internal flow dater *the dater* $x_Y^{\mathcal{F}}(w) = \max_{X \in \mathcal{F}(w)} \{\delta_w(X_0, Y_1) + \eta(X)\}$.

The final flow dater *is:* $y^{\mathcal{F}}(w) = \max_{Y \in \mathcal{F}(w)} \{x_Y^{\mathcal{F}}(w) + \sigma(Y)\}$

From $(\eta, \mathcal{O}, \sigma)$ we define a $(\max,+)$-automata, $Q^{\mathcal{F}}$, called *flow automata*. Its set of state is $\mathcal{F} = \bigcup_{a\in\Sigma} \mathcal{F}(a)$, its transitions are defined by the matrix $\mu(a)$ where $\mu(a)_{ij} = \delta_a(X_0^i, X_1^j)$ for X^i, X^j in $\mathcal{F}(a)$ and $-\infty$ elsewhere. The vector of initial (resp. final) condition is $\eta = (\eta(X^1), \dots, \eta(X^k))$ (resp. $\sigma = (\sigma(X^1), \dots, \sigma(X^k))$).

Proposition 1 *The automata $Q^{\mathcal{F}}$ recognizes $y^{\mathcal{F}}$, i.e the internal daters of $Q^{\mathcal{F}}$ are the internal flow daters and the final dater is the final flow dater. In others terms, if r is the power serie recognized by $Q^{\mathcal{F}}$, then $\langle r, w\rangle = y^{\mathcal{F}}(w)$, for all $w \in \Sigma^*$.*

From the logical point of view, we only want to select words recognized by the Σ-automata Q_Σ. Recall that the tensor product of two automata over a semiring is defined by the tensor product of their linear representations. As a boolean (classical) automata can be seen as a $(\max,+)$-automata, we have the

Definition 4 *The* timed diagram D *of an order automaton* $(Q_\Sigma, \eta, \mathcal{O}, \sigma)$ *is* $D =_{def} Q_\Sigma \otimes^t Q^{\mathcal{F}}$.

Note that if L is the language recognized by Q_Σ and r is the formal power serie recognized by $Q^{\mathcal{F}}$ then the $(\max,+)$-automata D recognizes the Hadamard product of the series: $r \odot_{\text{H}} car(L)$.

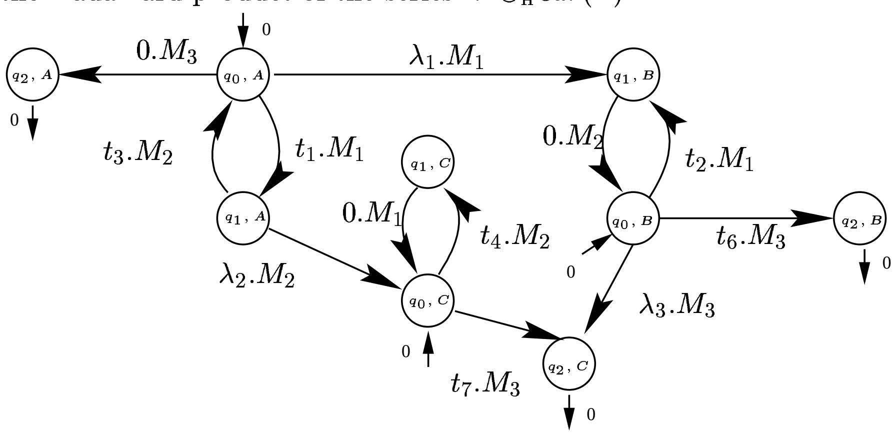


Figure 4 Timed diagram of the order automata of figure 3

For all $A \in \mathcal{F}$ and all $q \in Q_\Sigma$, the internal daters $x_{A,q}$ of D define formal series $s_{A,q} \in \mathbb{R}_{\max}[[\Sigma]]$ given by $\forall w \in \Sigma^*$, $\langle s_{A,q}, w\rangle = x_{A,q}(w)$. Moreover, they define formal series s_A by: $\langle s_A, w\rangle = \bigoplus_{q\in Q_\Sigma} \langle s_{A,q}, w\rangle.\sigma_q$. So, for all $w \in \Sigma^*$, the serie s recognized by the timed diagram verifies: $\langle s, w\rangle = \bigoplus_{A\in\mathcal{F}} \langle s_A, w\rangle.\ \beta_A$, where α (resp β) is the vector of initial (resp. final) condition in the linear representation of Q_Σ.

Let $A^{\mathcal{F}} \in \mathcal{M}_n(\mathbb{R}_{max})$ be the matrix of the automata $Q^{\mathcal{F}}$ and $A_\Sigma \in \mathcal{M}_m(\mathbb{R}_{max})$ the matrix of Q_Σ. The behavior of the timed diagram is given by the least solution of $\Xi = \Xi \, . \, (A^{\mathcal{F}} \otimes^{t} A_\Sigma) \oplus \eta \otimes^{t} \alpha$, where Ξ is in $\mathbb{R}_{\max}[[\Sigma]]^{nm}$ and verifies $\Xi_{ij} = s_{X_i,q_j}$ for $1 \leq i \leq n$ and $1 \leq j \leq m$. So, the behavior of the timed diagram gives the earliest behavior of the timed system described by the order automata.

4. FROM HMSC TO ORDER AUTOMATA

Let $H =< N, \longrightarrow, l, \mathcal{M}_t >$ be a timed HMSC. First, to each bMSC M of $\mathcal{M}_t$, $M = \langle E, \leq, A, I(M), \alpha, \Delta, T \rangle$ we associate a pondered bipartite order $\mathcal{O}(M)$:

We note $\mathcal{C}_{A\to B}$ the (finite) set of chains from A to B, i.e. the chains c such that $min(c) = min(E_{|A})$ and $max(c) = max(E_{|B})$. To any chain $c = e_1 \leq \cdots \leq e_n \in \mathcal{C}_{A\to B}$ we associate a delay:

$$d(c) = \Delta(e_1) + h(e_1, e_2) + \Delta(e_2) + ... + h(e_{n-1}, e_n) + \Delta(e_n)$$

where the application h is:

$$\begin{array}{llll} h: & E \times E & \longrightarrow & \mathbb{R}_+ \cup \{-\infty\} \\ & (e_1, e_2) & \mapsto & \begin{cases} T(e_1, e_2) & \text{if } \exists m \text{ a message with } e_1 = !m, e_2 = ?m \\ 0 & \text{if } \phi(e_1) = \phi(e_2) \\ -\infty & \text{else} \end{cases} \end{array}$$

The pondered bipartite order $\mathcal{O}(M)$ associated to a bMSC M can now be defined as a tuple $(I_0 \cup I_1, \leq_M, \delta_M)$, where:

- I_0 and I_1 are two disjoint copies of $I = \bigcup_{M\in\mathcal{M}_t} I(M)$,
- $\leq_M$ is the order $\{(X_0, Y_1) \in I_0 \times I_1, \mathcal{C}_{X\to Y} \neq \emptyset\} \cup \{(X_0, X_1) \in I_0 \times I_1, E_{|X} = \emptyset\}$,
- and δ_M is the map:

$$\begin{array}{llll} \delta_M: & I_0 \times I_1 & \longrightarrow & \mathbb{R}_+ \cup \{-\infty\} \\ & (X_0, Y_1) & \mapsto & \begin{cases} \max_{c\in\mathcal{C}_{X\to Y}} d(c) & \text{if } \mathcal{C}_{X\to Y} \neq \emptyset \\ 0 & \text{if } X = Y \text{ and } E_{|X} = \emptyset \\ -\infty & \text{else} \end{cases} \end{array}$$

Now, as the HMSC H has an input node $q_0 \in N$, and output nodes, $F \subset N$, we can define for H the order automata $(Q, \eta, \mathcal{O}, \sigma)$, where $Q = (N, \{\longrightarrow, l\}, \{q_0\}, F)$ is an automaton over the alphabet $\mathcal{M}_t$, with set of nodes N, input node $\{q_0\}$, set of output nodes F and $\{\longrightarrow, l\}$ the transitions. $\mathcal{O}$ is the previously defined map with domain $\mathcal{M}_t$. As there is no initial delay on the instances, α and β are such that $\forall A \in \bigcup_{M\in\mathcal{M}_t} I(M), \alpha_A = \beta_A = 0$.

Example: The HMSC of Figure 1 is transformed into the order automata Figure 3, with $a = M_1$, $b = M_2$, $c = M_3$, $t_1 = \delta_1$, $\lambda_1 = \delta_1.\tau_1.\delta_2$, $t_2 = \delta_2$, $t_3 = \delta_3.\delta_5$, $\lambda_2 = \delta_3.\tau_2.\delta_4$, $t_4 = \delta_4$, $t_6 = \delta_6$, $\lambda_3 = \delta_6.\tau_3.\delta_7$, $t_7 = \delta_7$ and, hence, gives us the timed diagram of Figure 4.

Proposition 2 *Let $W \in \mathcal{M}_t^*$ be a possible execution of a HMSC H. We note $d_X(W)$ the completion date on instance $X \in \bigcup_{M \in \mathcal{M}_t} I(M)$ of the sequence of bMSCs W. We call $d(W)$ the maximum on these dates. Then the timed diagram of H caracterizes its temporal behavior; i.e the internal daters recognize $d_X(W)$ and the final dater y verifies $y(W) = d(W)$.*

Proof: One proceed by recurrence on the lenght of W:

1) If $W = M$. Let $d(M)$ be the date of completion of M and, for a instance A, $d_A(M)$ be the completion date of M on A. The quantity $x_{q2,A}(M)$ is the maximal delay associated to chains of M witch terminate on A. All chains with $d(c) = x_{q2,A}(M)$ have $\max(A)$ as maximal element. MSC M is completed on A when $\max(A)$ terminates its execution. This execution is possible if and only if all the element preceding $\max(A)$ have been executed. As two incomparable elements are concurrent, the execution time of all the events $\leq \max(A)$ is the biggest delay of execution of the chains with maximal element $\max(A)$. Now M is completed when all the instances have finished their executions; the maximum of the $x_{q2,A}(M)$, i.e $y(M)$, gives us $d(M)$.

2) Suppose that, for $A \in I(M)$ and $|W| = n$, we have $x_{q,A}(W) = d_A(W)$; that is, by construction of the timed diagram, $d_A(W) = \bigoplus_{X \in I(W)} \delta_W(X_0, A_1)$. The sequential composition of W with M gives us:

$$d_A(W.M) = \bigoplus_{X \in I(W)} d_X(W). \max_{c \in \mathcal{C}^M_{X \to A}} d(C)$$

By the hypothese of reccurence we have:

$$\begin{aligned} d_A(W.M) &= \bigoplus_{X \in I(W)} \bigoplus_{Y \in I(M)} \delta_W(Y_0, X_1).\delta_M(X_0, A_1) \\ &= \bigoplus_{Y \in I(M)} \delta_W(Y_0, A_1) = x_{q,A}(W.M) \end{aligned}$$

5. A (max, +) STUDY OF TRAFFIC IN HMSC'S

Let $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ and S be the following linear system:

$$(S) \quad \begin{cases} x(k) = A \otimes x(k-1); \\ x(0) \in \mathbb{R}^n_{\max} \end{cases}$$

The asymptotic behavior of S is given by the cycle-time vector:

$$\chi(A) = \lim_{k \to +\infty} \frac{1}{k} \times x(k)$$

This vector exists when A has at least one finite entry per row and is independent of $x(0)$; it can be computed by the (max, +) policy improvement algorithm described in [8]. The maximum of its entries is the spectral radius $\rho_{max}(A)$ of A. The (max, +) Perron-Frobenius theorem tells us that it is equal to the maximal circuit mean of A:

$$\rho_{\max} = \max_{1\leq k\leq n} \max_{i_1,\ldots,i_k} \frac{A_{i_1 i_2} + \cdots + A_{i_k i_1}}{k}$$

This theorem is the cornerstone of the calculus of mean traffic in asymptotic behavior of a HMSC.

Proposition 3 *Let $U.V^\omega$ be an ultimately periodic execution of a HMSC H, with $U = N_1 \ldots N_p, V = M_1 \ldots M_n \in \mathcal{M}_t^*$, let N_V be the number of messages contained in V, n the length of the word V and μ the linear representation of the timed diagram D associated to H. Then the asymptotic mean traffic t_m is:*

$$t_m = \frac{N_V}{n.\rho_{max}(\mu(M_1)...\mu(M_n))}$$

Proof: First, notice that the asymptotic mean traffic only depends on the specification defined by V. Second, if another HMSC admits $U.V^\omega$ as a possible execution, then it should have the same t_m. So suppose that $N = N_1; \ldots; N_p$ and $M = M_1; \ldots; M_n$ are the concatenations of the partials orders defined by the N_is ($i \in 1..p$) and the M_js ($j \in 1..n$). If μ' is the linear representation of the timed diagram of this new HMSC, say H'; then, because of the proportionality of the maximum mean circuit, we have $n.\rho_{max}(\mu(M_1)...\mu(M_n)) = \rho_{max}(\mu'(M))$. For H', proposition 2 implies $\rho_{max}(\mu'(M)) = \lim_{k\to+\infty} \frac{y(M^k)}{k}$, which is clearly the time needed to execute M in stationary rate. As the number of messages in M is equal to N_V, and by definition of t_m, we have $\rho_{max}(\mu'(M)) = \frac{N_V}{t_m}$, which concludes the proof. □

The local and global durations of a sequence, and the delay needed before returning into a given state are provided by the daters of the time diagrams. Delaying a process just consists in modifying the initial conditions of the system. Traffic t_m gives information on the asymptotic temporal behavior of a HMSC, which can be refined by the study of the cycle-time vector and by the spectrum of the timed diagram. Consider, for example, Figure 2: in the first case ($\delta_1 < \delta_2$), the cycle time depends only on δ_2, while in the second case ($\delta_1 > \delta_2$), it depends on both values. This provides information on processes idle time.

6. CONCLUSION

In this paper, we have defined a (max,+) approach for studying performance from high level specifications of distributed systems. First timed Message Sequence Charts are transformed in order automata. Then a (max,+) analysis allows for studying the temporal behavior of the specification.

References

[1] H. Ben-Abdallah and S. Leue. Timing Constraints in Message Sequence Chart Specifications. Proceedings of the Tenth Conference on Formal Description Techniques FORTE/PSTV'97, Osaka, Japan, Nov. 1997, Chapman & Hall.

[2] N. Faltin, L. Lambert, A. Mitschele-Thiel, F. Slomka. PMSC – Integrating Performance into Message Sequence Chart. Formale Beschreibungstechniken für verteilte Systeme, 7. GI/ITG-Fachgespräch, Berlin, Juni 1997.

[3] S.Gaubert, Performance evaluation of (max,+) automata. IEEE transactions on automatic Control, 40(12), Dec 1995.

[4] A. Benveniste, C. Jard, S. Gaubert, "Algebraic techniques for timed systems". In 9th International Conference on Concurrency Theory, Concur'98, Nice, France, sep. 1998.

[5] P.Kosiuczenko Formalizing Time aspects of Message Sequence Charts. Technical Report Nr. 9703, Ludwig-Maximilians-Universität München, Institut für Informatik, January 1997.

[6] V.Pratt Modeling Concurrency with Partial Orders, International Journal of Parallel Programming, Vol 15, No 1, 1986, pp 33-71

[7] F. Slomka, J. Zant, L. Lambert. MSC-based Schedulability Analysis. Workshop on Performance and Time in SDL and MSC. Technical Report 1/98, IMMD VII, University of Erlangen-Nuremberg, Erlangen, February 1998.

[8] J. Cochet-Terrasson, G. Cohen, S. Gaubert, M. Mc Gettrick, J.P. Quadrat,Numerical computation of spectral elements in max-plus algebra, IFAC Conference on System Structure and Control, 1998

[9] ITU, Norm Z.120, Message Sequence Charts.