Some Pathological Message Sequence Charts, and How to Detect Them

Loïc Hélouët

France Télécom R&D,
2 avenue Pierre Marzin, 22307 Lannion Cedex, France,
loic.helouet@francetelecom.com

Abstract. Some confusing Message Sequence Charts are identified, that can be considered as syntactically correct, but may lead to ambiguous interpretations. The first kind of MSC identified appears when parallel components of a parallel frame synchronize implicitly to continue an execution. The second case is called non-local choice, and appears when more than one instance is responsible for a choice. Non-local choice has already been studied before. This paper provides an extension of the definitions and corresponding detection algorithms. The third case is confluent MSCs, and appears when concurrency is expressed through a choice.

1 Introduction

Since the first standard appeared in 1992, Message Sequence Charts have gained a lot of expressive power. Many elements have been added to the original language: composition in MSC-96, additional time measurement possibilities, and variables in MSC-2000. All these improvements are obviously aiming at a better usability of Message Sequence Charts, and are mainly driven by expressed needs. However, adding a new element to a language may fully satisfy some users, and introduce at the same time confusion for all the others, or even semantics ambiguities.

The main elements of the language (instances, messages, timers) are usually well understood. Furthermore, as far as basic Message Sequence Charts are concerned, very few ambiguities can arise. It is not true when considering composition through parallel, choice, or loop operator, or through High-level Message Sequence Charts. Some graphical inconsistencies in MSC'96 were already pointed out by [7].

HMSCs, for example, allow for the definition of MSCs that are considered as syntactically correct, but the intuitive understanding of which are different from the behavior allowed by the semantics. In most cases, these HMSCs should be considered as pathological, and rejected. Fortunately, the cases introduced within this paper can be easily detected. This detection is based on global properties of the MSC.

As a correct syntax does not ensure the correct understanding of an MSC specification, we argue that a new document should be added to the appendices

R. Reed and J. Reed (Eds.): SDL 2001, LNCS 2078, pp. 348–364, 2001.

of recommendation Z.120, as a "methodological guideline", which should define what a valid MSC should be. The identification of at least two pathological cases and an extension of the definition of non local choice is a first contribution in this direction.

This paper is organized as follows: first we quickly recall the operational semantics of High-level Message Sequence Charts, as defined by [11,8]. Then, Sect. 3 shows the first ambiguous case, that arises when two parallel components synchronize without any communication. Section 4 recalls the definition of non-local choice, proposes an algorithm to detect it, and then discusses its pathological character. Section 5identifies another pathological kind of MSC, and proposes an algorithm to detect it.

2 Operational Semantics

2.1 Basic Message Sequence Charts

Many semantics have been proposed for basic Message Sequence Charts (bMSC). The semantics retained for recommendation Z.120 is based on process algebra. However, we consider as natural to model bMSC as a finite, labeled partial order, as in [1,6].

A bMSC is a tuple $M = (E, \leq, A, I, \alpha)$ where:

- E is a finite set of events,
- $\leq$ is a partial order relation (antisymmetric, reflexive and transitive) called *causal order* on events,
- I is a set of names of instances that perform at least one action in M, and is called the set of *active instances* of M.
- A is a set of action names.
- $\alpha : E \longrightarrow A \times I$ is a labeling of events.

From now, we will note $\phi(e)$ the instance performing event e, ie the instance $i \in I$ such that $\alpha(e) = (a, i)$ for some $a \in A$. Slightly abusing the notation, we will note $\phi(E) = \{\phi(e) | e \in E\}$ the set of instances appearing in any set of events E. For any MSC M, we will note $min(M) = \{e \in E \mid \nexists\, e' \neq e \text{ and } e' \leq e\}$ the set of minimal events of M. For any labeled order $M = (E_M, \leq_M, A_M, I_M, \alpha_M)$, for any set $E' \subseteq E_M$, we will denote by $M_{/E'}$ the restriction of M to events of E'. We will also denote by $M_\emptyset =< \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >$ the empty MSC.

A bMSC defined with a partial order model has the same semantics as the process algebra definition of [11]. The main difference is that we use one single operational semantics rule :

$$\frac{e \in min(M), \alpha(e) = (a, i)}{M =< E_M, \leq_M, A_M, I_M, \alpha_M > \quad \xrightarrow{a} \quad M_{E-\{e\}}}$$