Interpretation of DD-LOTOS Specification by C-DATA*

Maarouk Toufik Messaoud[1](✉), Saidouni Djamel Eddine[2], Mahdaoui Rafik[1], and Houassi Hichem[1]

[1] Faculty of ST, ICOSI Lab, University Khenchela, BP 1252 EL Houria, 40004 Khenchela, Algeria
tmaarouk@gmail.com, {mehdaoui.rafik,houassi_h}@yahoo.fr

[2] Faculty of NTIC, MISC Lab, Univ Constantine 2, Constantine, Algeria
saidouni@hotmail.com

Abstract. The DD-LOTOS language is defined for the formal specification of distributed real-time systems. The peculiarity of this language compared to existing languages is its taken into account of the distributed aspect of real-time systems. DD-LOTOS has been defined on a semantic model of true concurrency ie the semantics of maximality. Our work focuses on the translation of DD-LOTOS specifications to an adequate semantic model. The destination model is a communicating timed automaton with durations of actions, temporal constraints and supports communication between localities; this model is called C-DATA*.

Keywords: Real time system · Duration of actions · Formal specification · DD-LOTOS language · C-DATA*

1 Introduction

The specification of real-time systems is very important step, these systems are everywhere in our environment. Moreover, they are often part of critical systems in various fields, such as aviation, industrial process control, control of nuclear power plants, etc.

Formal methods play a fundamental role in the various stages of the process engineering of computer systems especially in real-time systems. Formal methods allows to specify a critical system using formalisms, langanges and models defined on a formal semantics. The essential feature in this approach is unambiguously specify critical systems in the different phases of its life cycle, and to validate a number of its requirements. In recent years the proposal of models, languages and formalism defined on a formellle semantics, able to specify critical real-time systems have known a lot of progress. Most research focuses on the extension of existing models and in particular on the process algebra. Among these works focus on the extension of LOTOS[3][6][4][5][10]. The process algebra is a formal framework for the specification and analysis of complex systems in general, and in particular real-time systems.

T. Morzy et al. (Eds): ADBIS 2015, CCIS 539, pp. 414–423, 2015.
DOI: 10.1007/978-3-319-23201-0_42

Our aim is to design an environment for the formal specification and compilation of concurrent systems. The specification is described in formal language DD-LOTOS, this language allows to describe distributed systems with temporal constraints, then the specification is translated into the semantic model C-DATA*[7] to be verified by the formal verification tools.

2 Distributed D-LOTOS Language

2.1 Syntax

The DD-LOTOS[7] language represents an extension of D-LOTOS language to support the distribution and communication between the localities, it was enriched with the following features:

- The explicit distribution,
- Remote communication.

Distribution is ensured by introducing the notion of locality. Localities exchange information by the message exchange paradigm. The syntax of DD-LOTOS is defined as follows:

$$
\begin{array}{rl}
E ::= & \textbf{Behaviors} \\
 & \mathit{stop} \mid \mathit{exit}\{d\} \mid \Delta^d E \mid X[L] \mid \\
 & g@t[SP]; E \mid i@t\{d\}; E \mid \mathit{hide}\, L\, \mathit{in}\, E \mid \\
 & E[]E \mid E|[L]|E \mid \ E \gg E \mid E\,[> E \\
 & a!v\{d\}; E \\
 & a?xE \\
S ::= & \textbf{Systems} \\
 & \phi \mid \ \ S \mid S \ \mid \ l(E)
\end{array}
$$

Fig. 1. *Syntax of DD-LOTOS*

Let PN, ranged over by $X, Y ...$, be an infinite set of process identifiers, and let $\mathcal{G}$, ranged over by g, set of gates (observable actions). $i \notin \mathcal{G}$ is the internal action and $\delta \notin \mathcal{G}$ is the successful termination action. $\mathit{Act} = \mathcal{G} \cup \{i, \delta\}$, ranged over by α, is the set of actions. L denotes any finite subset of $\mathcal{G}$. The terms of DD-LOTOS are named behavior expressions, $\mathcal{B}$ ranged over by $E, F, ...$ denotes set of behavior expressions.

Let $\mathcal{D}$ be a domain of time. $\tau : \mathit{Act} \rightarrow \mathcal{D}$ is the duration function which associates to each action its duration. We assume $\tau(i) = \tau(\delta) = 0$. Let g be an action, E a behavior expression and $d \in \mathcal{D}$ a value in the temporal domain.

The main syntax concerns the syntax of systems S and behavior expression E. The informal semantics of syntactic items is the following:

- Informally $a\{d\}$ means that action a has to begin its execution in a temporal interval $[0, d]$. $\Delta^d E$ means that no evolution of E is allowed before the end of a delay equal to d. In $g@t[SP]; E$ (resp. $i@t\{d\}; E$) t is a temporal variable recording the time taken after the sensitization of the action g (resp. i) and which will be substituted by zero when this action ends its execution.
- The basic operators of process algebras as: nondeterministic choice $E[]E$, parallel composition $E|[L]|E$, the interiorization $hide\, L\, in\, E$, sequential composition $E \gg E$, and preemption $E\,[> E$.
- The expression $a!v\{d\}; E$, specifies the emission message v via the communication channel a. This emission operation must occur in the temporal interval $[0, d]$.
- On the other side, the behavior expression $a?xE$ specifies the message receiving on channel a. The received message substitutes the variable value x. This variable is used in the behavior expression E.
- A system may be either:
  - Empty, expressed by ϕ,
  - The composition of sub systems $S \mid S$, or
  - A behavior expression E in a locality l expressed by $l(E)$.

Definition 1. *(actions) The actions in global system are:*

- Set of communication actions between localities: are emission or receiving messages through a communication channel $Act_{com} ::= a!m\ |a?x\ |\tau$ (output actions, input actions and the silent action).
- Set $Act = \mathcal{G} \cup \{i, \delta\}$ previously defined.

Definition 2. *(Localities and channels): The set $\mathcal{L}$ ranged over by l, denotes set of localities. ϑ an infinite set of channels defined by users ranged over by a,b,... channels are used for communication message between localities.*

2.2 Structured Operational Semantics

The operational semantics of behaviors are given by the operational semantics of D-LOTOS. This semantic is extended to DD-LOTOS by giving the semantics rules for communicated systems as follows:

Process $a!v\{d\}; E$: Let us consider the configuration ${}_M[a!v\{d\}; E]$, the emission of the message v begins once the actions indexed by the set M have finished their execution, conditioned by the condition $Wait(M)$ which must be equal to $false$ in rule 1. Rules 2 and 3 express the fact that the time attached to the process of sends cannot begin to elapse until all the actions referenced by M are finished. Rule 4 imposes that the occurrence of the action of sends takes place for the period d, otherwise the process is transformed to $Stop$.

1. $$\frac{\neg Wait(M)}{{}_M[a!v\{d\};E] \overset{M^{a!v}\,x}{\longrightarrow} {}_{\{x:a!v:t\}}[E]} \qquad x = get(\mathcal{M})$$

2. $$\frac{Wait(M^{d'})\ or\ (\neg Wait(M^{d'})\ and\ \forall \varepsilon{>}0.\ Wait(M^{d'-\varepsilon})) \qquad d'{>}0}{{}_M[a!v\{d\};E] \xrightarrow{d'} {}_{M^{d'}}[a!v\{d\};E]}$$

3. $\dfrac{\neg Wait(M)}{{}_M[a!v\{d'+d\};E] \xrightarrow{d} {}_M[a!v\{d'\};E]}$

4. $\dfrac{\neg Wait(M)\ and\ d'>d}{{}_M[a!v\{d\};E] \xrightarrow{d'} {}_M[stop]}$

Process $a?xE$: Let us consider the configuration ${}_M[a?xE]$, the following rule expresses that the receiving starts once the action indexed by set M have finished their execution.

$$\frac{\neg Wait(M)}{{}_M[a?xE] \overset{M^{a?x}\,y}{\longrightarrow} {}_{\{y:a?x:0\}}[E]}$$

Remote Communication

Distributed activities exchange messages between them, the expression $l(a!v\{d\})$, expresses that the message v is offered for a duration d. By an activity at the locality l, the message v should be sent on channel a. On the other side, $k(a?xE)$ specify that activity E in locality k is ready to receive a message on channel a. The following rule defines the remote communication between two distributed activities via the channel a. In this case communication will be specified by silent (i) evolution as follows: (action silencieuse τ:

$$\frac{-}{{}_M[l(a!v\{d\};E1)]|_{M'}[k(a?xE2)] \xrightarrow{\tau} {}_M[l(E1)]|_{M'}[k(E2\{v/x\})]}$$

Time Evolution on System

$$\frac{E \xrightarrow{d} E'}{l(E) \xrightarrow{d} l(E')}$$

$$\frac{S_1 \xrightarrow{d} S_1' \quad S_2 \xrightarrow{d} S_2'}{S_1 \mid S_2 \xrightarrow{d} S_1' \mid S_2'}$$

3 Distributed Semantic Model for Distributed Realtime Systems

The behavior of a real system can be represented by a transition system under certain assumptions of abstraction. The formalism of timed automata (TA's) was introduced by Rajeev Alur and David Dill in [1]. Its definition provides a simple way to provide transitions systems a set of temporal constraints expressed using real variables called clocks. The model of timed automata is constructed in conformity with the hypothesis of structural and temporal atomicity actions.

The model Durational Action Timed Automata (DATA)[2] is introduced with the aim of take into account the explicit durations of action, an extension of this approach to systems with constrained time in order to take into account the temporal constraints and the urgency of action, this model is called DATA*[9]. The objective of this section is the presentation of a semantic models that can support the distributed aspect of real-time distributed systems.

3.1 Durational Action Timed Automata(DATA*)

In this section, we describe a method for taking into account of non-atomicity temporal and structural of actions in timed automata, through the DATA model. In general, temporal constrained systems can not be completely specified if we do not consider concepts such urgency, deadlines, constraints, etc. To account for these concepts, the DATA* model's was defined in[9].

Let H, ranged over by $x, y...$ be a set of clocks with non-negative (in a time domain T as Q^+ or R^+). The set $\Phi_t(H)$ of temporal constraints γ Over H, is defined by the syntax $\gamma ::= x \sim t$, where x is a clock in H, $\sim \in \{=, <, >, \leq, \geq\}$ and $t \in Q^+$. A DATA * A is a quintuplet (S, L_S, s_0, H, T) tel que:

1. S is a finite set of states, and
2. $L_S : S \rightarrow 2^{\Phi_t(H)}_{fn}$ is a function which corresponds to each state s the set F of ending conditions(duration conditions) of actions possibly in execution in s,
3. $s_0 \in S$ is the initial state,
4. H is a finite set of clocks, and
5. $T \subseteq S \times 2^{\Phi_t(H)}_{fn} \times 2^{\Phi_t(H)}_{fn} \times Act \times H \times S$ is the set of transitions. A transition (s, G, D, a, x, s) represents switch from state s to state s, by starting execution of action a and resetting clock x. G is the corresponding guard which must be satisfied to fire this transition. D is the corresponding deadline which requires, at the moment of its satisfaction, that action a must occur. (s, G, D, a, x, s) can be written $s \xrightarrow{G,D,a,x} s'$.

The semantics of a DATA* $A = (S, L_S, s_0, H, T)$ is defined by associating to it an infinite transitions system S_A over $Act \bigcup T$. A state of S_A (or configuration) is a pair $< s, v >$ such as s is a state of A and v is a valuation for H. A configuration $< s_0, v_0 >$ is initial if s_0 is the initial state of A and $\forall x \in H$, $v_0(x) = 0$. Two types of transitions between S_A configurations are possible, and which correspond respectively to time passing and launching of a transition from A.

Example of DATA * is given in figure 2.

3.2 Communicating Durational Action Timed Automata (C-DATA)

C-DATA*[7] is a semantic model that allows taking into account of all the aspects present in the DATA* s model, such as non-atomicity temporal and structural of the actions, urgency of the actions, deadlines and temporal constraints. In the C-DATA* each locality is represented by a DATA*, the global system is represented by the set of DATA* s locals, which communicate by exchanging messages through communication channels

Definition 3. *A Communicating DATA (C-DATA) $A(S, L_S, s_0, \vartheta, H, \Pi, T_D)$ represents a subsystem with:*

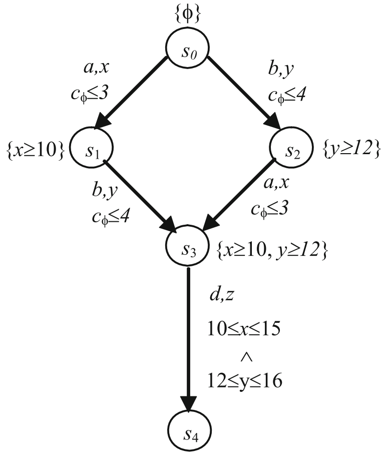

Fig. 2. Example of DATA*

- *S is a finite set of states,*
- *$L_S : S \to 2^{\Phi_t(H)}_{fn}$ is a function which corresponds to each state s the set F of ending conditions(duration conditions) of actions possibly in execution in s,*
- *$s_0 \in S$ is the initial state,*
- *ϑ is the alphabet of the channels on which messages flow between the subsystems.*
- *H is a finite set of clocks,*
- *$\Pi = Act_{com} \cup Act$, is the set of internal and communication actions of A, and*
- *$T_D \subseteq S \times 2^{\Phi_t(H)}_{fn} \times 2^{\Phi_t(H)}_{fn} \times \Pi \times H \times S$ is the set of transitions.*
  A transition $(s, G, D, \alpha/(a(!/?)v)/i, z, s\prime)$ represents switch from state s to state $s\prime$, by starting execution of action $\alpha \in Act$ or actions (Sending or Receiving) or synchronization for the accomplishment of communication (silent action) and updating clock z.
  G is the corresponding guard which must be satisfied to fire this transition.
  D is the corresponding deadline which requires, at the moment of its satisfaction, that action α must occur.
  $(s, G, D, \alpha/(a(!/?)v)/\tau, z, s\prime)$ can be written
  $s \xrightarrow{G\ ,\ D\ ,\ alpha\ /\ (\ a\ (\ !/\ ?\ )v)\ /i\ ),\ z} s'$.

Definition 4. *1. **System:** A system of n C-DATA is a tuple $S = (A_1, \ldots, A_n)$, with $A_i = (S_i, L_{S_i}, s_{0_i}, \vartheta, H_i, \Pi_i, T_{iD})$ a C-DATA.*

*2. **States:** $GS(S) = (s_1, v_1) \times \ldots \times (s_n, v_n) \times (\vartheta^*)^p$, is The set of states.*

*3. **Initial State:** The initial state of S is:$q_0 = ((s_{01}, 0), \ldots, (s_{0n}, 0) : \epsilon_1, \ldots, \epsilon_p)$ such as ϵ is the empty word on the alphabet ϑ*

*4. **System States:** Let $S = (A_1, \ldots, A_n)$ a system of n C-DATA, $A_i = (S_i, L_{S_i}, s_{0_i}, \vartheta, H_i, \Pi_i, T_{iD})$:*
A global state of S is defined by the state of each subsystem and the states

of each channel, a state of S is an element of $(s_1, v_1) \times \ldots \times (s_n, v_n) \times (\vartheta^*)^p$ *such that* $v_i(h)$ *are valuations on* H.

4 Interpretation of DD-LOTOS Specifications to C-DATA*

This section is devoted to the implementation of our specification environment.

4.1 System Architecture

The system receives as input a specification of real-time behavior expressed in DD-LOTS. This specification must be checked lexically and syntactically before generating the C-DATA* matching. The generation of the C-DATA* is from AST (Abstract Tree Synrtax) generated by the parser. The architecture of our system is as follows:

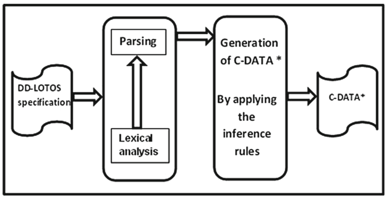

Fig. 3. System architecture

4.2 The Analysis Steps

From the syntax of the DD-LOTOS, we define the grammar below: the terminals are bold, non-terminals in italics. The axiom of the grammar is the non-terminal SPECIFICTION.

DD-LOTOS Specification:

$< specification >::= SYSTEM < entete >:=< code > ENDSYS$

$< entete >::= id[< params >]$

$< params >::= id < last\_params > |id[nombre] < last\_params > |epsilon$

$< last\_params >::=, id < last\_params > |, id[nombre] < last\_params > |epsilon$

$< code >::=< expression >< where\_exp > |epsilon$

$< where\_exp >::= WHERE < decl\_proc >< other\_decl > |epsilon$

Process:

$< processus >::= PROCESS < entete >:=< code > ENDPROC$

$< other\_proc >::=< processus >< other\_proc > |epsilon$

Behaviors:

$< expression >::= STOP|EXIT\{Duree\}|NIL$

$| < expression >< opr >< expression >$
$|HIDE < hide\_params > IN < expression >$
$|(< expression >)$
$| < expression >< para - op >< expression >$
$|DELAY < Duree >< expression >$
$|G[< params >]$
$| < act >; < expression >$
$|G@ < Entier > [SP]; < expression >$
$|i@ < Entier > \{Duree\}; < expression >$
$< hide\_params >::= id < last\_params>$
And grammar of operators, identifiers, and durations of actions.

4.3 Representation of a C-DATA*

The C-DATA* will be represented by a graph whose nodes represent the states and the edges represent transitions. An abstract syntax tree AST nodes and leaves.

4.4 AST Transformation Algorithm to a Graph

This algorithm transforms an AST into a graph.
Input: AST.
Output: Graph that represents the list of configurations representing the initial specification.
Begin
While there are lines to visit in the AST Do

- Apply the corresponding operational rule, calls the implementation procedure of each rule,
- Extract the resulting configuration,
- Skip to the next line.

End While
End

4.5 Example (sender2receivers)

In this section we will try to apply our application on example of sender2receivers[8]. DD-LOTOS specification of this example is as follows:

Specification $sender2receivers[a, b]$
Behavior
$l(E)|k(P)|n(Q)$.

Where
Process $E ::= E1|||E2$
Where

$E1 ::= (a!v\{5\}|b!v\{5\}) >> DELAY 5 E1$
$E2 ::= (c?xB) >> E2$
Endproc
Process $P ::= (a?xB[]DELAY 5; c!nack\{5\}) >> P$
Endproc
Process $Q ::= (b?xB[]DELAY 5; c!nack\{5\}) >> Q$
Endproc
Endspec

Once the compilation completes successfully, we generates the corresponding C-DATA*.

The execution of this example is shown in the figure 4:

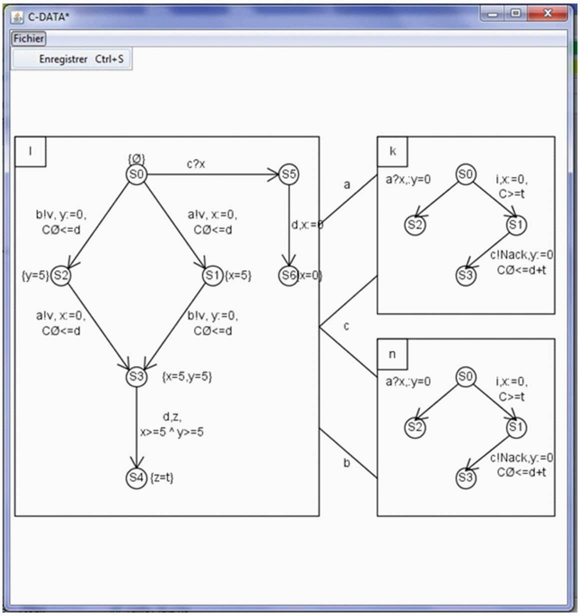

Fig. 4. Final C-DATA*

5 Conclusion

In this paper we presented a contribution to the specification of real-time distributed systems, with explicit duration of actions. In a previous work[7], we introduced the notion of locality required for modeling the distributed aspect of distributed systems.

The main interest of our approach is the proposal of a language defined on true concurrency semantics: semantics of maximality[10] which allows the explicit expression of durations, and it supports temporal constraints including urgency of actions. Concerning communication, we have defined local and remote communication, when two processes want to communicate, so are on the same locality, then the communication is ensured through the gates which are defined locally. If both processes are on two different localities then the message exchange is the way of communication.

References

1. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science **126**, 183–235 (1994). Elsevier
2. Belala, N., Saïdouni, D.E.: Non-Atomicity in Timed Models, International Arab Conference on Information Technology, Al-Isra Private University, Jordan, LIRE Laboratory, University of Mentouri, 25000 Constantine, Algeria (December 2005)
3. Bolognesi, T., Lucidi, F.: LOTOS-like process algebras with urgent or timed interactions. In: Parker, K.R., Rose, G.A. (eds.) FORTE. IFIP Transactions, vol. C-2, pp. 249–264. North-Holland (1991)
4. Courtiat, J.P., de Oliveira, R.C.: On RT-LOTOS and its Application to the Formal Design of Multimedia Protocols. Annals of Telecommunications **50**, 11–12 (1995)
5. Courtiat, J.P., Santos, C.A.S., Lohr, C., Outtaj, B.: Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique. Computer Communications **23**, 1104–1123 (2000). Elsevier
6. Léonard, L., Leduc, G.: A Formal Definition of Time in LOTOS - Extended Abstract. Formal Aspects of Computing **10**, 248–266 (1998). BCS
7. Maarouk, T.M., Saïdouni, D.E., Khergag, M.: DD-LOTOS: a distributed real time language. In: Proceedings 2nd Annual International Conference on Advances in Distributed and Parallel Computing, Special Track: Real Time and Embedded Systems, pp. 45–50. Singapore (2011)
8. Maarouk, T.M.: Modèles formels pour la conception des systèmes temps réel, thèse de Doctorat. Laboratoire MISC, Constantine, Algérie (2012)
9. Saïdouni, D.E., Belala, N.: Actions duration in timed models. In: International Arab Conference on Information Technology. Yarmouk University, Irbid, December 2006
10. Saïdouni, D.E., Courtiat, J.P.: Prise En Compte Des Durées D'action Dans Les Algèbres deProcessus Par L'utilisation de la Sémantique de Maximalité. Ingénierie des Protocoles, Hermes (2003)