Comments on Comments on "An Interval Logic for Real-Time System Specification"

Pierfrancesco Bellini, Paolo Nesi, Davide Rogai

***Abstract*—The paper on Comments on "An Interval Logic for Real-Time System Specification" presents some remarks on the comparison examples from TILCO and other logics and some slips on the related examples. This paper gives evidence that such issues have no impact on the validity of the TILCO Theory of paper [1] and provides some further clarifications about some aspects of the comparison.**

I. Introduction

The paper "An Interval Logic for Real-Time System Specification" authored by R. Mattolini and P. Nesi (IEEE Transaction on Software Engineering, March 2001) [1] was submitted in 1996, revised in 1997 and finally accepted in June 1998. Paper [1] reported solid theoretical results about TILCO temporal logic.

This paper is meant to be a response to the paper contained in this Transactions issue and titled "*Comments on "An Interval Logic for Real Time System Specification""*, by C.A. Furia, A. Morzenti, M. Pradella and G. Rossi, hereafter referred to as "Paper on Comments on [1]" or **[C1]** in short. [C1] paper has provided some comments which are related on a very small part of the paper [1], and more precisely they are limited to some examples and not to the TILCO theory. In [C1], there are no passages as likely to invalidate the TILCO Theory; on the other hand it offers corrections on minor examples about the usage of the TILCO formalism for the specification production, when compared to other formalisms.

This paper aims at: (i) giving evidence that the issues reported in [C1] have no impact both on the validity of the TILCO Theory and of paper [1], (ii) clarifying some statements reported in [C1] that are devious and not clear regarding the usability of TILCO for effective specifications.

This paper is structured as follows. Section II reports facts on TILCO theory and clarifies the marginal relevance of [C1] in general terms and with respect to the issues proposed in [1]. Section III provides explanation for some devious and uncalled for statements in [C1], on the TILCO logic usability and readability, including comments on the comparison of TILCO with TRIO.

Manuscript received ……………. This work was partially supported by the …………………… .

P. Bellini is with the University of Florence, Department of Systems and Informatics, Florence, Italy (email: pbellini@dsi.unifi.it)

P. Nesi is with the University of Florence, Department of Systems and Informatics, Via S. Marta 3, 50239 Florence, Italy (corresponding author, phone +39-055-4796523, fax: +39-055-4796363, email: nesi@dsi.unifi.it)

D. Rogai is with the University of Florence, Department of Systems and Informatics, Florence, Italy (email: rogai@dsi.unifi.it)

II. Facts and TILCO theory

In [1], the TILCO temporal logic for the specification, validation and verification of real-time systems was presented. TILCO is a first order temporal logic providing (i) a metric for time (thus allowing a specification of qualitative and quantitative timing constraints), (ii) decidability for a wide set of formulas (non-temporal quantification must bind only variables with types over finite domains), and (iii) no explicit quantification over the temporal domain. For TILCO, a sound axiomatization was proposed and then used to build a deductive system in natural deduction style. This has been used to prove various TILCO theorems. As to our experience, being TILCO based on FOL (First Order Logic) and an implicit model of time, it turned out to be particularly suitable for requirements analysis and incremental specification of real-time systems. This is also due to the fact that TILCO supports the validation activity during all phases of the system life-cycle by means of its formalization in the automatic theorem prover Isabelle/HOL [12] and recently also in PVS [11] This allows the validation for refinement and the proof of general system properties.

Furthermore, the main goal of the whole TILCO development in the several years is the effective execution of the specifications, which is achieved by using the TILCO Executor [4], [5], [14]. TILCO executable model can also be used for operational validation [11].

The TILCO paper [1] consisted in 19 pages, where the authors brought forth:

- the theory of TILCO temporal logic with its definitions;
- the syntax in terms of BNF formalization;
- the TILCO semantics with explaining examples;
- the axiomatization of TILCO, by means of the FOL axioms and additional TILCO axioms for completeness and consistency, while presenting also inference rules for introduction and elimination, other general rules, and selected theorems and corollaries with their demonstrations;
- the discussion about the soundness and decidability of TILCO;
- the discussion about the support of proof of properties and executability of TILCO formulas and specifications;
- a complete specification example: the alternating bit protocol which includes several TILCO formulas and their related validation, and finally;
- a comparison with other temporal logics.

Therefore, paper [1] on TILCO reported solid theoretical results about TILCO temporal logic and its usage, which has not been criticized nor discussed in [C1].

The TILCO logics and theory reported in [1] have been further developed so as to create more powerful formalisms; such as TILCO-X [3] and C-TILCO [2]. TILCO-X extends TILCO introducing the new syntax and semantics of **@** and **?** operators (dynamic intervals and bounded happen). This extension allows both removing **since** and **until** TILCO operators and writing simple predicates, including counting of events that may occur in intervals [3]. CTILCO supports process composition and decomposition by allowing the specification of communicating TILCO processes [2].

A. Some corrections on examples of Section 5.1

Section 5.1 of [1] reported a comparison between TILCO and mainly two other temporal logics: namely TRIO [8], [10] and MTL [13].. The largest part of [C1] is focused on some mistakes which occurred when producing some formulas in Section 5 of [1]. The TILCO examples used for the comparison in Section 5.1 are not consistent with the rest of the TILCO paper. Those examples are limited to only a page of the 19 ones paper [1] is made of.

In Section 5.1 of [1], some of the TRIO formulas were not written as a very TRIO author would prefer, while in other cases there are some slips like the flip of parameters of temporal operators. The authors of [1] were probably misled by the availability of many user defined temporal operators, which happens often in TRIO.

For such reasons, we would like to thank the authors of [C1] for the corrections they provided.

At the same time for formula (3a) of [C1], it is true that the formula does not consider a double request of resource reservation, on the other hand, it is easy to avoid this problem in the usual situation when the requester, after issuing a request, is put in a waiting for the grant status. We acknowledge this reasoning to be more suitable when the specification is aiming at its execution, rather than at its validation, and this is the main emphasis we have always given to the whole TILCO approach. This fact has not been considered at all by the authors of [C1].

B. TILCO theory

We would like to remark again that the theoretical results presented in [1] about TILCO temporal logic have neither been criticized, nor brought up for discussion in the Paper on Comments on [1], [C1].

III. Comparison of Logics

The second part of [C1] criticized the comparison between TILCO and TRIO made in [1]. The intention is clearly to defend TRIO. In such critics, its authors accuse the authors of [1] of "*…..chose to ignore the Dist() operator*" of TRIO in favor of Futr() and Past() operators of TRIO. In order to object to this statement, firstly we would like to bring the discussion back to its context that grew on the basis of the issues of readability and conciseness of temporal logics.

A. Readability and conciseness

In Section 5 of [1], a discussion in cognitive terms about readability and conciseness of temporal logics was reported. That discussion apparently satisfied the reviewers, but not the very TRIO authors, or at least not after 4-5 years have gone. The discussion was mainly on the assessment of the cognitive readability and conciseness of temporal logics on the basis of the number of operators used in certain formulas.

In [1], what has been remarked is as follows: on the basis of some examples, typical specifications produced in TILCO used fewer distinct operators than what occurred with the same specifications provided in TRIO. The difference on the number of operators used in the two logics depended on many factors. The aim of the TILCO logic definition was to provide a small number of operators with their maximum power without losing readability, conciseness and executability of specifications [1].

B. The TRIO logic and Dist() operator

The understanding of TRIO logic by the authors of [1] at the time they wrote the article (1994-96) was based upon the relevant literature published at that time, where neither the Dist() operator appeared, or it was merely mentioned as a basic operator to give the semantics of more "complex" operators, such as Futr(), Past(), Alw(), etc., and not to be used in the specifications based on TRIO. On the basis of those operators TRIO presented a distinction from past and future in the specifications like other temporal logics. These facts are self-evident out of papers dated 1994-96 by the very authors of TRIO: [6], [7], [8] and [9] in which Dist() operator is not mentioned at all, and the basic operators are Futr() and Past() operators, and in paper [10] where Dist() is used only in the semantic and not for the specification (please note that a large part of these references are those mentioned in [C1]). Therefore, as to the period when research of [1] was produced, Dist() operator was never used by the very authors of TRIO for the production of specification (it was a clear message for the readers of those papers) and it was supposed to be part of the semantics definition, as used in [10]. The Dist() operator has been gradually introduced after 1996, whereas in all previous papers of TRIO, the logic was presented as only based on Futr() and Past() operators. Even in papers after 1996 and in many recent papers, the Dist() operator is used only for the definition of other higher level operators.

The massive usage of Dist() operator could indeed bring the users to specify the temporal constraints in substantially FOL plus Dist(). This is hard to be accepted in terms of usability and it is the reason why many researchers have defined other temporal logics operators gaining conciseness/readability and obviously restricting the expressive power of FOL and it is one of the reasons for the usage by the very TRIO authors of other higher level operators for the specification.

The authors of the [C1] accused the authors of [1] having intentionally neglected the presence of Dist() operator. This was not the case for several reasons:

- Dist() operator was also mentioned in [1];
- the authors of [1] have only taken examples as provided in many papers written by the very authors of TRIO in which the operator Dist() has never been used for the specification as stated above;
- the usage of Dist() operator is not improving the readability of TRIO as discussed hereafter.

In many cases, as in the philosophy of TRIO, it is better to use higher level operators, while this is left to the users as performed by the authors of [1]. The authors of TILCO paper have seen the Dist() operator and considered it as it was considered by the very authors of TRIO: a special function for the specification of semantics, used in some versions of TRIO, but not in all of them, and not to be used for the specifications. Therefore it could not be used in the comparison between TRIO and TILCO formalisms by using their typical specifications.

C. Some Examples

In Table 1 of this paper, the specifications of some typical constraints produced for TILCO and TRIO (based on Dist()) are reported. The examples below have been proposed in order to provide an objective measure on cognitive complexity and conciseness,. The conciseness is to a certain extent a measure of the power of a language in a given context. Conciseness can be defined as the ratio between the "*cognitive complexity of a given model of a concept in a certain formalism*" and the "*complexity of such a concept*". On such grounds, in a comparison based on the modelling of the same concepts, it is enough to compare the cognitive complexity of the different models obtained by the different formalisms, as in Table 1.

Concept	TILCO	TRIO
Alw. Past	$A\,@(-\infty,0)$	$\forall t(t<0 \rightarrow \mathbf{Dist}(A,t))$
Alw. Future	$A\,@(0,+\infty)$	$\forall t(t>0 \rightarrow \mathbf{Dist}(A,t))$
Always	$A\,@(-\infty,+\infty)$	$\forall t(\mathbf{Dist}(A,t))$
Since Weak	$\mathbf{since}(B,A)$	$\begin{array}{l}\forall t''(t''<0 \rightarrow \mathbf{Dist}(A,t''))\vee \\ \exists t\begin{pmatrix} t<0 \wedge \mathbf{Dist}(B,t)\wedge \\ \forall t'(t<t'<0 \rightarrow \mathbf{Dist}(A,t'))\end{pmatrix}\end{array}$
Until Weak	$\mathbf{until}(B,A)$	$\begin{array}{l}\forall t''(t''>0 \rightarrow \mathbf{Dist}(A,t''))\vee \\ \exists t\begin{pmatrix} t>0 \wedge \mathbf{Dist}(B,t)\wedge \\ \forall t'(0<t'<t \rightarrow \mathbf{Dist}(A,t'))\end{pmatrix}\end{array}$
Lasts	$A\,@(0,t)$	$\forall t'(0<t'<t \rightarrow \mathbf{Dist}(A,t'))$
Lasted	$A\,@(-t,0)$	$\forall t'(-t<t'<0 \rightarrow \mathbf{Dist}(A,t'))$
Within Past	$A\,?(-t,0)$	$\exists t'(-t<t'<0 \wedge \mathbf{Dist}(A,t'))$
Within Future	$A\,?(0,t)$	$\exists t'(0<t'<t \wedge \mathbf{Dist}(A,t'))$
Within	$A\,?(-t_1,t_2)$	$\exists t(-t_1<t<t_2 \wedge \mathbf{Dist}(A,t))$
Was	$A\,@(-t_1,-t_2)$	$\forall t(-t_1<t<-t_2 \rightarrow \mathbf{Dist}(A,t))$
Will be	$A\,@(t_1,t_2)$	$\forall t(t_1<t<t_2 \rightarrow \mathbf{Dist}(A,t))$
Could be	$A\,?(t_1,t_2)$	$\exists t(t_1<t<t_2 \wedge \mathbf{Dist}(A,t))$

Table 1 – Specification of simple constraints in TILCO and TRIO.

It becomes clear how TILCO operators subsume the Dist() operator, together with a quantification over an interval, as already stated in [1]. We can let the readers draw a conclusion on which of the logic is more readable and/or concise and how many operators are used in the specification of the two logics, where for operators the reader could consider: <, >, ∀, ∃, @, ?, (,), [,], -, →, ∧, ∨, ∞, etc. For the sake of completeness the comparison with Table 3 of [1] can be performed to see if the specifications based on Futr() and Past() operators of TRIO resulted more or less complex to understand/concise with respect to the corresponding ones in TILCO or in TRIO with Dist(). When focusing on the comparison of the two tables, it is evident that the number of operators used in both TRIO cases is practically the same, apart from the cases where the temporal constraint includes both past and future (for example for Within() operator). In those cases, the TRIO specifications with Dist() resulted simpler than those with Past() and Futr().

As to the operators, it should be noted that the TILCO paper claims the "basic" nature of TILCO operator '@' and '?' because they are the sole operators able to express predicates on intervals as a built-in feature of the TILCO language. In TRIO, any operator can be defined as a function, but it is a different way of creating a logic formalism. TILCO defines **@** and **?** to deal with time intervals and **since** and **until** to deal with ordering of predicates. Please note that, with TILCO-X Dynamic intervals **since** and **until** are represented by @ operator with an enhanced semantic [3] as reported in Table 2. Also in this case the differences between TILCO-X and TRIO are evident.

Concept	Specification
Since Weak	TILCO-X: $A\,@(-B,0)$ TRIO: see Table 1
Until Weak	TILCO-X: $A\,@(0,+B)$ TRIO: see Table 1
If a sequence Q-R* occurs, then P occurs at least 2 times between Q and R (* Q is the first occurrence after another Q-R sequence or since the beginning, R is the first occurrence after Q)	TILCO-X: $\begin{pmatrix}\begin{pmatrix} Q \wedge \neg Q\,@(-R,0)\wedge \\ R\,?(0,+\infty)\end{pmatrix}\rightarrow \\ P\,?_2(0,+R)\end{pmatrix}@(-\infty,+\infty)$ TRIO: $\forall t_Q \forall t_R \left[\begin{array}{l}\begin{pmatrix} t_Q<t_R \wedge \mathbf{Dist}(Q,t_Q)\wedge \mathbf{Dist}(R,t_R)\wedge \\ \left(\begin{array}{l}\exists t' \begin{pmatrix} t'<t_Q \wedge \mathbf{Dist}(R,t')\wedge \\ \forall t''\begin{pmatrix} t'<t''<t_Q \rightarrow \\ \mathbf{Dist}(\neg R \wedge \neg Q,t'')\end{pmatrix}\end{pmatrix}\vee \\ \forall t''(t''<t_A \rightarrow \neg\mathbf{Dist}(Q,t''))\end{array}\right)\end{pmatrix}\rightarrow \\ \exists t_1\exists t_2\begin{pmatrix} t_Q<t_1<t_2<t_R \wedge \\ \mathbf{Dist}(P,t_1)\wedge \mathbf{Dist}(P,t_2)\end{pmatrix}\end{array}\right]$
From now on, S will respond to P before R	TILCO-X: $(P \rightarrow S\,?(0,+R))@(0,+\infty)$ TRIO: $\forall t\left[\begin{array}{l}(t>0 \wedge \mathbf{Dist}(P,t))\rightarrow \\ \exists t'\begin{pmatrix} t'>t \wedge \mathbf{Dist}(S,t')\wedge \\ \forall t''(t<t''<t')\rightarrow \neg\mathbf{Dist}(R,t'')\end{pmatrix}\end{array}\right]$

Table 2 – Specification of typical patterns in TILCO-X and TRIO.

The above discussion does not mean that TILCO as a temporal logic is more expressive than TRIO, that is FOL plus a Dist() operator. FOL has a greater expressive power, but it is intrinsically not decidable. On the other hand, many temporal logics derived from FOL have been created to simplify reasoning on time improving readability and decidability with respect to the direct usage of FOL. In particular, TILCO has been designed to simplify the readability and the executability at the expense of expressiveness. On the other hand, for the execution of TILCO specifications [3], [4], decidability is needed. In TRIO you can define a specific new operator to express any kind of formula. Obviously this does not mean that the readability is higher, since those new operators have to be understood to read the formula.

On this regard, it is more interesting to make a comparison of TILCO and the TRIO subset providing decidability as reported in [15] of 2003. In that case, the semantic of TRIO has been provided in terms of Dist() while the very authors of TRIO provided a number of different operators such as Since(), Until(), Futr(), Past(), Lasts(), and Lasted() for the specifications and avoiding the quantifications over time dependent variables as in TILCO. We prefer to leave to the readers the production of such comparison as a useful exercise

D. *Strong or week since and until*

We think that the selection of "strong" or "weak" *since* and *until* operators is not a good argument in order to compare temporal logics. This is totally different from what has been reported in [C1] Section III. It is obvious that if the weak ones are built-in, the strong ones have to be expressed by using more complex formulas and vice versa. What is to be investigated is which of these operators are mostly used in the specification activity: if the typical constraints consider a "weak" nature, then the ability of expressing this with basic operators makes the formulae more concise. In TILCO, the decision of using the weak versions was also motivated by the aim of having a simpler executable semantics for operators managing event sequences.

E. *Minimality*

The authors of [C1] stated that the basic operators in the logics should be chosen to pursue minimality of definition and economy of reasoning in the proof of the meta-theorems of the logic, not to obtain conciseness or readability in the specification. It is correct that a low number of operators in a logic may help in working on theorems, while a reduced set of operators to be used in the specification brings forth conciseness or readability in the specifications. The approach used in TRIO is based on the proliferation of operators since each user may define specific own operators, and the used FOL model leaves more expressive power at the expense of using explicit quantification over time. All these aspects lead to a high degree of fragmentation, which forces the readers to analyze more terms in order to understand the single temporal constraint. It is a different model of writing temporal constraint with respect to TILCO or other interval logics.

IV. Conclusions

In this paper, we have commented [C1] containing comments on [1], in the effort of giving evidence that (i) the issues reported in [C1] have no impact on the validity of the TILCO Theory and of [1], (ii) some further explanations about some devious and not clear statements reported in [C1] regarding the usability of TILCO for effective specifications.

In the comparison of TILCO and TRIO, even considering the Dist() operator as suggested by the very TRIO authors, the specification which can be obtained in several examples resulted to be strongly structurally different than the one in TILCO. More specifically, in this paper the specifications of TRIO, based on Dist() turned out to be very similar to those provided in [1] with Futr() and Past() operators, except for concepts in which time intervals including both past and future were involved. It is noteworthy that, when comparing Table 1 of this paper with Table 3 of [1], the specifications based on TRIO use a larger number of operators than the equivalent specification in TILCO. Therefore, we believe that they forced the reader to a greater effort in understanding and producing the specification as already stated in [1].

As a conclusion, we would like to thank again the authors of [C1] for pointing out some inaccuracy in [1], but we want to state again that TILCO has its own value, proved within the period of many years of experience. The principles lying behind the TILCO are based on different grounds, which is a different understanding of readability and conciseness, as well as a strong interest in the executability of the logic.

We believe that the distinctive nature of TILCO can attract practical users, and we invite the readers to experiment the logic and its tools with the extended features, which are available at http://www.dsi.unifi.it/~tilco

Acknowledgment

We wish to acknowledge A. Fantechi for his helpful suggestions on the issues discussed in this paper.

References

[1] R. Mattolini, P. Nesi, “An Interval Logic for Real-Time System Specification”, IEEE Transactions on Software Engineering, Vol.27, N.3, pp208-227, 2001

[2] P. Bellini, P. Nesi, “Communicating TILCO: a Model for Real-Time System Specification. Proc. of the 7th IEEE Int. Conf. on Engineering of Complex Computer Systems, ICECCS 2001, IEEE Press, Skovde, Sweden}. (June 2001), pp.4-14.

[3] P. Bellini, P. Nesi, “TILCO-X an Extension of TILCO Temporal Logic”, Proc. of the 7th IEEE Int. Conf. on Engineering of Complex Computer Systems, ICECCS 2001, IEEE Press, Skovde, Sweden. (June 2001), pp.15-25.

[4] P. Bellini, A. Giotti, P. Nesi, “Execution of temporal logic specifications” Proc. of the 8th IEEE Int. Conf. on Engineering of Complex Computer Systems, ICECCS 2002, IEEE Press, GreenBelt, Maryland. (Dec. 2002), pp.78-88.

[5] P. Bellini, A. Giotti, P. Nesi, D. Rogai “TILCO Temporal Logic for Real-Time Systems Implementation in C++”, Proc. of the 15th Int. Conf. on Software engineering and knowledge engineering, SEKE03, ACM press, San Francisco Bay. (Jun. 2003).

[6] A. Gargantini, L. Liberati, A. Morzenti, C. Zacchetti, “Specifying, validating, and testing a traffic management system in the TRIO environment”, Computer Assurance, 1996. COMPASS '96, 'Systems Integrity. Software Safety. Process Security'. Proceedings of the Eleventh Annual Conference on, pp.65–76, 17-21 June 1996.

[7] A. Morzenti, P. San Pietro, “Object-oriented logical specification of time-critical systems”, ACM Transactions on Software Engineering and Methodology (TOSEM), Vol.3, n.1, January 1994,

[8] M. Felder, A. Morzenti, "Validating real-time systems by history-checking TRIO specifications", ACM Transactions on Software Engineering and Methodology (TOSEM), Vol.3, n.4, October 1994

[9] D. Mandrioli, S. Morasca, A. Morzenti, "Generating test cases for real-time systems from logic specifications", ACM Transactions on Computer Systems (TOCS), Vol.13, n.4, Nov. 1995.

[10] M. Felder, D. Mandrioli, A. Morzenti, "Proving properties of real-time systems through logical specifications and Petri net models, IEEE Transactions on Software Engineering, Vol.20, n., pp.127-141, 1994

[11] D. Rogai, "Hybrid approach to real-time system development", Ph.D. Thesis, University of Florence, 2006.

[12] P. Bellini, "Interval Temporal Logic for Real-Time Systems: Specification, Execution and Verification Processes", Ph.D. Thesis, University of Florence, 2001.

[13] R. Koymans, "Specifying real-time properties with metric temporal logic", Journal of Real-time System, Kluwer Academic Publishers, Norwell, MA, USA Vol.2 , n.4 , pp.255-299, 1990

[14] TILCO Web site, http://www.dsi.unifi.it/~tilco

[15] A. Morzenti, M. Pradella, P. San Pietro, P. Spoletini, "Model-checking TRIO specifications in SPIN", 12th International Formal Methods Symposium (FM2003), LNCS 2805, Pisa, Italy - September 8-14, 2003, pp. 542-561