I will present now some reflections about the verification of temporal constraints for control command systems by means of Petri nets.

Here is the composition of the group developing this work
CURRENT STATE OF THE ART

When the word verification is associated with Petri nets, the first thing that comes to mind is the analysis with respect to the so-called "good properties" which are: bounded, live and reversible. These "good" properties ensure a kind of consistency and completeness for the model of the control command system.

Then there is the invariant analysis. p-invariants and t-invariants are specific sub-nets which ensure some specific properties for the model. p-invariants are mainly used to easily prove that some mutual exclusion constraints are guaranteed in any situation by the control command system. It is therefore possible to prove that the disjunctive resources are not simultaneously allocated to two or more users.

These two approaches are now well known.

A third point is that it should be clear that starting from a Petri net with an initial marking, it is possible to generate the reachability graph i.e. the set of all the reachable states, and therefore to use all the results of automata based proofs, or temporal logic ones. Many speakers, expert in the domain, have already presented these techniques during the workshop. I do not think that the fact that the automata is derived from a Petri net is very important.

An important point for me is that for people working in industry, the main validation is frequently done by means of informal reasonings which are directly developed on the paths of the Petri nets.

The aim of this presentation is to try to understand why these reasonings seem more cogent than some formal proofs and to try to formalize these reasonings.
T 2 TRUTH AND TIME

The issue is the verification of temporal constraints. This means that the propositions or better the logical formulas which have to be handled are true with respect to some time intervals.

If the relationships between the truth of a proposition and the time are analyzed, at least three cases can be differentiated:

- eternal truths
- transitory truths
- consumable resources

Let us explore in detail these three categories
T 3 TRUTH AND TIME

Eternal truths are the classical case.

If you have the proposition P1 "human are mortal" and the proposition P2 "Socrates is human", then you can derive that the proposition P3 "Socrates is mortal".

You have this kind of inference scheme. The same scheme is obtained if the propositions are equations or inaquations such as $x < y$, $z$ is strictly greater than $y$ and it is derived that $z$ is strictly greater than $x$.

Two states can be pointed out during the reasoning. Initially P1 and P2 were known to be true. At the end of the reasoning, P1, P2 and P3 are known to be true. The monotony property which implies that a proposition which has been proved to be true cannot be proved to be false is always verified.

Within the context of the verification of control command systems specified by means of Petri nets, propositions which are eternal truths are required. Indeed it is necessary to prove that some propositions are true for any marking, in any situation.
T 4 TRUTH AND TIME

Let us know consider another kind of propositions.

Now proposition P1 is "the weather is fine", proposition P2 is "I am in Montréal". I may deduce that I visit the city which is proposition P3.

I have the same logical scheme as before. Assume now that a storm breaks out. This will result in aborting my visit.

In this new scheme 3 states can be defined. In the first one propositions P1 and P2 were true. In the second one P1, P2 and P3 were true. But in the third one, just after the storm, only proposition P1 reamains true.

Monotony is verified in the states S1 and S2. If we restrict to proposition P2, it is verified for S1, S2 and S3. But monotony is not verified for P1 and P3 between S2 and S3.

This kind of propositions is required in the context of control command systems when you want to prove that something is true for some specific marking M. The major difficulty is that sometimes monotony does hold and sometimes not.
Let us now consider the last case. Proposition P1 is "machine m is idle", proposition P2 is "part p is ready to be machined". The derived proposition is P3 "part p is machined on m".

The derivation scheme seems the same as before but now, at the exact moment of the derivation, machine m is no longer idle and part p no longer ready to be machined. In a way the logical propositions P1 and P2 have been consumed in order to produce P3.

If the knowledge states are examined, In the state S1 propositions P1 and P2 are known to be true and in the state S2 only proposition P3 is known to be true.

If only propositions of this kind are handled, monotony is never verified. One may have the feeling that this kind of situation is simpler than the preceding one because the "behavior" of the proposition is always the same. However as it contradicts monotonicity, it is better to handle these propositions differently as eternal truths.

When the verification of control command systems is involved, this kind of proposition is very frequently used. Indeed the notion of resources which are allocated (and therefore consumed) and released (therefore produced) is essential. In a Petri net, each time an informal reasoning based on the token player is developed, it is based on consumable resources: the tokens. The propositions are mainly: what are the tokens required in order to cover some path i.e. to execute some firing sequence?
T 6 Resources as logical propositions

Let us now analyze the consequences of the fact of considering propositions which are systematically consumed when they are used.

The first one is that this implies the possibility of counting the propositions.

Indeed, if it makes sense to state "machine m is idle", it also makes sense to state that "machine m and m prime are idle" i.e. that "2 machines are idle". This could be denoted 2.P1.

If in addition we have the proposition P2 "part p is ready" then we may deduce that p is machined on one of the machines (proposition P3) and that one machine remains idle (proposition P1).

This means that the inference scheme P1, P2 entails P3 is transformed into 2.P1, P2 entails P1, P3 which differs from what could be obtained by using the monotonicity i.e. P1, P2 entails P1, P2, P3. Linearity i.e. the fact that if you add a proposition on the left part of the implication you have to add it on the right part also has to replace monotonicity i.e. the fact that any deduction remains valid if more propositions are known to be true.

Consuming is denoted by the operation minus one. This means that consuming implies counting.

This could be used as a criteria in order to characterize the kind of propositions which can be considered as consumable resources. It makes no sense to consider the proposition "2 humans are mortal" in place of "humans are mortal" or it would have a completely different meaning. It also does not make sense to write "2 weathers are fine" or "2 I am in Montréal".

One difficulty in making a bridge between Petri nets and logic is precisely the necessity of counting. If a proposition "Pi" is attached to a place "Pi", the proposition is generally considered false if the place is empty and true if the place contains one token. But what about a place containing 2 tokens? How can it be interpreted in classical logic?
T 7 Resources as logical propositions

What about the negation? What could be the negation of "1 machine is idle"

It could be "0 machine" but this proposition would be the same for any resource: "0 machine" is the same as "0 part". (In linear logic it is denoted by "1" the neutral element of TIMES connectives)
It could be "2 machines are idle" but what about 3, 4 and so on? "2 machines" is denoted by 2.P1
The only interpretation which is consistent with the notion of resources which are consumed and produced is "I need one idle machine"

Among the consequences of this is the fact that P1 and not(P1) are not contradictory. Indeed it is possible to have one machine free and to need another one, for example because 2 parts have to be machined

As a direct consequence of this fact, the notion of classical negation is lost with eternal truth. If there are no eternal truths, there is no "absurdity" or eternal contradiction. Negation just transforms consuming into producing and vice versa. "1 idle machine has been produced and is currently available to be consumed" is transformed into "it is needed to produce one idle machine which would be eventually consumed".

In the case of control command systems, this corresponds to reversing the time. It transforms a point of view "decision" (which operation should be performed in the next future) into a point of view "diagnosis" (by considering past evolutions how can I explain a present situation).

On a Petri net model, it transforms the net into its dual (i.e. the net obtained by reversing all the arcs).
T 8 Resources as logical propositions

Let us now consider the issue of decision making representation. This fragment of Petri net may be interpreted as the fact that an idle machine (a token in place a) can be allocated either to operation "b" or to operation "c", but not to the two operations simultaneously.

In classical logic, from a implies b and from a implies c it can be derived that from a I can deduce b and c. This formula is valid in classical logic but it does not depict the problem. Without any time considerations it is true that, as long as the resource can be re-used, it will be possible to execute the operations "b" and "c”. But these operations will have to be executed in sequence and the situation in which "b” and "c are executed simultaneously is only possible if 2 machines are idle (i.e. 2 tokens in a).

It is then required to operate within the frame of a non-classical logic. Girard's linear logic is well-suited. Within the frame of this new logic, it is possible to deduce that if a linealy implies b and if a linearly implies c, then it is possible to deduce that from 2a it is possible to obtain b TIMES c that is a state for which b and c are simultaneously consumable. But the formula a linearly implies b TIMES c cannot be deduced. Actually the number of a on the left part of the sequent symbol (turnstile) and on its right part does not balanced.

In the framework of Girard's linear logic i.e. in the framework of a logic that only handles propositions which are consumable resources, it is possible to correctly represent a decision within a valid formula i.e. with a unique logical world. Within the frame of classical logic, it is necessary to consider 3 worlds, each one being consistent with monotonicity. The world for which a is true, the one corresponding to b and the one to c.
T 9 Resources as logical propositions

Let us summarize the motivations for defining a new logic. J.Y. Girard has defined the linear logic in order to be consistent with the notion of resources.

He started from the sequent calculus because it is the natural way for using logic to describe reachability issues under the form of knowledge transformations.

In contrast with some attempts in Artificial Intelligence which try to solve the frame problem by adding new rules, J.Y; Girard has just suppressed the two rules (weakening and contraction) which were directly connected to monotonicity.

This new logic is very consistent with many informal reasonings and observations of any Petri net addict. For instance, as long as the Petri net describes actions corresponding to the consuming and producing of tokens, its structure remains simple and easy to analyse. What is dreadful is the introduction of arcs with the purpose of testing some token counts of places without modifying them. For example elementary loops and inhibitor arcs. The structure of the net turns complex, the analysis is often impossible and in some case has been proved not decidable (inhibitor arcs). All these primitives actually correspond to an introduction of classical logic propositions within a framework of consumable resources. It is not sound to try to consume and produce eternal truth in the same way that it is not sound to transform consumable resources in eternal truths.
Let us refine the relationships between Petri nets and linear logic. More precisely, the need is a translation of a Petri net model into a set of linear logic formulas.

The net structure is described, transitions by transitions. Actually a transition is associated with a linear logic implication. Let us consider the fragment of Petri net here:

- **t1**: a token in a and a token in d are simultaneously consumed in order to produce a token in c
- **t2**: a token in b and a token in c are simultaneously consumed in order to simultaneously produce a token in e and a token in f
- **t3**: a token in e is consumed to simultaneously produce a token in d and a token in g

The derivation of a firing sequence from the formulas attached to the transitions is described by a sequent (which has to be valid by construction). If transitions t1, t2 and t3 are fired in sequence and with this order then the resulting marking transformation will be:

One token simultaneously consumed in a b and d and one token produced in f, g and d

The tokens in c and e do not appear because it is certain that they will be produced before being consumed. The presence of b is required before firing the whole sequence because we want to ensure that the whole sequence will be fired until the end when it is started.

The derivation of the characterization of a sequence is a natural deduction rule in linear logic which is directly derived from the cut rule of the linear logic sequent calculus.

The effective firing of a sequence (or of a transition) is represented by sequents of the form: ...

It means that if the sequence is fired from marking M1, then marking M2 is obtained.

Such sequents have to be valid in linear logic. Its consistency with the Petri net structure is ensured by the fact that only linear implications denoting some transitions may be used for deriving the sequents characterizing the sequences. The fact that a sequent is valid is an eternal truth. Actually it represents a potential evolution. If the current marking is M1 and if we decide to fire the sequence, then the obtained marking will be M2. A sequent may be used as a proposition in a classical logic reasoning.
T 11 Petri nets and linear logic

An important point to underline is the fact that although linear logic is commutative in the same way as classical logic, the sequent which characterizes the sequence depends on the firing order.

For the sequence $t_1 \ t_2 \ t_3$ we have ...
For the sequence $t_2 \ t_3 \ t_1$ we have ...

The reason for which the sequents are different is the following one. In the first case the token in $c$ is produced before being consumed and it does not appear. In the second case it is the case of the token in $d$. In contrast, in the first case $d$ is consumed before being produced and so it appears in the formula and in the second case $c$ is consumed before being produced and it appears in the formula. The logic is commutative but the cut rule is not.

The characterization of the sequences in this formalism is therefore a generalization of the matrices $Pre$ and $Post$ for the transitions. This characterization is much more precise than the marking transformation which is usually considered.

The interesting point is that this characterization is a natural consequences of the structural rules defining linear logic. It derives from the absence of inverse in the monoids associated with the connectives ($P_1$ and not($P_1$) cannot be simplified).
Let us detail the way the sequent characterizing a sequence can be used. Consider the sequent representing the firing of sequence \(t_1 \ t_2 \ t_3\) on the net fragment. For the sequent \(\ldots\) to be valid it is necessary and sufficient to have \(M_1 = \ldots\) and \(M_2 = \ldots\).
Indeed it is necessary to balance the number of times each atom appear on the left and right side of the turnstyle. It is sufficient because we only have elements of the monoid in TIMES.

C is the context. It is a monomial in TIMES. It represents all the tokens which are not concerned by the firing of the sequence.

This means the requirement that the sequent is valid, implies constraints on the markings. From a sequence characterization we derive a constraint on the markings which can be used in any classical logic reasoning.

A necessary condition for the possibility of the firing of the sequence is that \(M_1\) and \(M_2\) are reachable markings. A sufficient condition is that \(M_1\) is a reachable marking.

It is important to point out that the sequent derivation and the constraints on \(M_1\) and \(M_2\) are obtained from the structure of the Petri net. They are independent from the initial marking and therefore it is not necessary to generate the reachability graph. The initial marking, and possibly the reachability graphs are only required when we use the sequent in a classical reasoning.
Is it possible to introduce explicit duration in this approach?

Let us consider the sequence t1 t2 and t3. It is necessary to wait for the completion of the firing of t1 and the production of a token in c in order to fire t2. In the same way it is necessary to wait for the completion of the firing of t2 and the production of a token in e in order to fire t3. This necessity directly derives from the way the sequent has been contructed.

Therefore the minimal duration of the sequence is directly the sum of the duration of each transition.

.....

.....

Once more this duration is totally independent of the possible locations of other tokens in the net and even of the fact that these tokens are used to fire other transitions. The sequent allows to exactly characterize the behavior which is been studied.
T 14 Petri nets and linear logic

The major advantage of this approach is the fact that true concurrency can be taken into consideration very easily.

Exactly in the same way a sequence can be characterized, it is possible to characterize specific scenarios which correspond to a set of firing sequences which are such that all of them start from the same initial marking and all of them ends for the same final markings.

Let us consider this example:

The sequence t1 t2 is characterized by:.....
The sequence t3 t4 is characterized by:.....
The trace resulting from any interleaving between these two sequences is simply characterized by: ..... which just results from the concatenation of the pre and the post of the two sub-sequences.

As represented in the Petri net fragment, there is actually true concurrency between the two sub-sequences because they do not share any token. The duration of the first sub-sequence is THETA ...
The duration of the second is .... and that of the trace is ...

We have the minimal duration of the trace without bothering of the exact "sequence" (interleaving) which will actually occur.
T 15 Conclusion

The approach which has been briefly introduced here is based on the use of linear logic in order to formalize intuitive reasonings which are commonly developed on Petri nets.

Its main characteristics are that the reasonings are:
1) not based on state enumeration, there is no need to construct the reachability graph of the Petri net
2) independent from the initial marking
3) based on the concept of sequences - the logic formulas describe state transformations, not properties of states

Item 2 is very important in control-command systems. For example it is not possible to verify the correctness of the control of a manufacturing system if the proof depends on the exact number of parts currently located in the shop. The proof has to be independent from this number.

This is the benefit. What is the cost?

The cost is that the propositions which can be handled only express reachability. It is possible to transform M1 into M2. Necessity cannot be handled directly. Necessity requires an enumeration of all the possibilities. In narrow relation with this, there is no negation in linear logic. No proof by contradiction is possible. For these reasons it is generally necessary to begin with a linear logic calculus and then to pass in classical logic by considering that a valid sequent is an eternal truth.