Offline methods of conformance testing for Antescofo

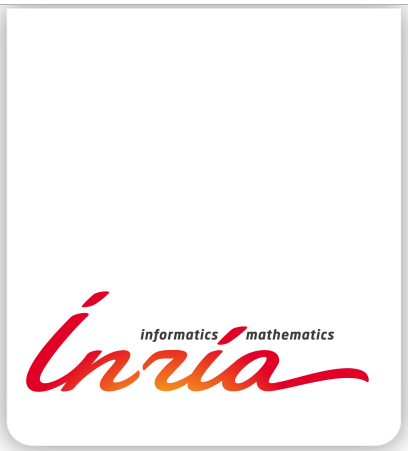
Clément Poncelet, Florent Jacquemard

Offline methods of conformance testing for Antescofo

Clement Poncelet, Florent Jacquemard

RESEARCH REPORT
N° 8700
March 2015
Project-Team MUTANT

ISSN 0249-6399 ISRN INRIA/RR--8700--FR+ENG



Offline methods of conformance testing for Antescofo

Clement Poncelet, Florent Jacquemard

Project-Team MUTANT

Research Report n° 8700 — March 2015 — 18 pages

Abstract: This report presents procedures for testing Antescofo, a score-based Interactive Music System (IMS). It describes common Model-Based Test (MBT) vocabulary and presents methodology and use cases.

Key-words: Model-based testing, offline, Antescofo tests, manual

RESEARCH CENTRE
PARIS – ROCQUENCOURT

Domaine de Voluceau, - Rocquencourt
B.P. 105 - 78153 Le Chesnay Cedex

Méthodes différées de conformité pour tester Antescofo

Résumé : Ce rapport présente des procédures de test pour Antescofo, un système interactions musicales (IMS) basé sur partition. Il décrit le vocabulaire courant des tests basés sur model (MBT) ainsi que sa méthodologie et ses cas d'applications.

Mots-clés : test basé sur model, différé, tester Antescofo

1 Model Based Testing

Figure 1 depicts in its upper half a reactive system's Implementation Under Test (IUT) interacting with an environment RealENV, and in its lower half, two formal specifications of the latter, resp. $\mathcal{S}$ and $\mathcal{E}$.

The *environment model* $\mathcal{E}$ is a formal description of what can be expected from the environment. In our case, it is the definition of the set of all possible input traces, *i.e.* all the potential interpretations of musicians to be tested on a given mixed-score.

Note that since IMS are realtime systems, we need to express time in $\mathcal{E}$ and $\mathcal{S}$, like in: *"one message m has to be emitted one beat after the first event* e_1 *of the musician"*.

We present the application of model based testing (MBT) techniques to a score based IMS called Antescofo. Roughly, our method proceeds with the following steps, depicted in Figure 2. First, a given mixed score is compiled into an intermediate representation (IR). This formalism can be presented as a table of finite state machines extended with delays and asynchronous communications with the environment (using input and output symbols) and between machines. It is viewed as the model $\mathcal{M}$ describing Antescofo reactions according to the score. Note that the model contains the environment ($\mathcal{E}$) and the system ($\mathcal{S}$) specifications.

The IR is then transformed into a timed automata (TA) network, assuming some restrictions. TA's are processed by tools from the Uppaal suite to generate test cases or expected output suites. In particular we use CoVer in some scenarios to make covering suites of test cases: input timed traces together with the corresponding expected output.

The input traces are then sent to the IUT (as a virtual musician's performance of the score) and the real outcome of the IUT is analyzed, resulting in a test verdict.

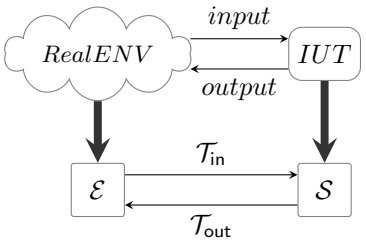


Figure 1: Specification : reality (top) and models (down)

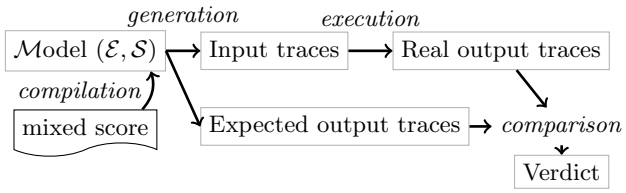


Figure 2: Score-based IMS testing workflow

2 Traces

2.1 Antescofo test representation

2.1.1 Architecture

Figure 3 describes roughly the architecture of Antescofo, which is made of two main modules. A *listening machine* (LM) decodes an audio or midi stream incoming from a musician and infers in realtime: (*i*) the musician's position in the given mixed score, (*ii*) the musician's instantaneous pace (*tempo*, in beats per minute). These values are sent to a *reactive engine* (RE) which schedules the electronic actions to be played, as specified in the mixed score. The actions are messages emitted *on time* to an audio environment. Therefore, the information exchanged between LM and RE as well as between RE and the output environment of the system is made of discrete events.

2.1.2 IO and Design Specific Language representation

Let $\mathcal{O}$ be a set of *output messages* (also called *action symbols* and denoted a) which can be emitted by the system and let $\mathcal{I}$ be a set of *event symbols* (denoted e) to be detected by the LM (i.e. positions in score).

In our Antescofo's DSL representation, an *action* is a term $\mathsf{act}(d, s, al)$ where d is the delay before starting the action, s is either an atom in $\mathcal{O}$ or a finite sequence of actions (such a sequence is called a *group*), and al is a list of attributes.

A *mixed score* is a finite sequence of *input events* of the form $\mathsf{evt}(e, d, s)$ where $e \in \mathcal{I}$, d is a duration and s is the top-level group *triggered* by e. Sequences

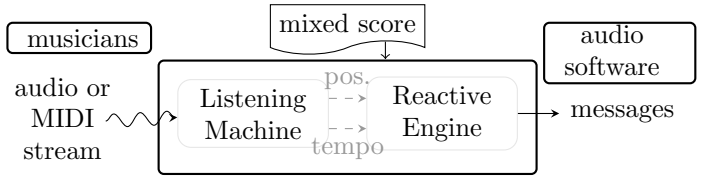


Figure 3: Architecture of Antescofo

are denoted with square brackets [,] and the empty sequence is [ ]. To resume, Antescofo DSL is interpreted with the grammar Figure 4.

We consider here two time units for expressing delays and durations d: (i) the number of beats (default unit): a logical time unit traditionally used in music scores that we call *relative time*, and (ii) milliseconds (ms), referred to as *physical time*.

The reconciliation of the relative and physical times is done through the detected tempo values. We consider a black-box testing conformance approach for Antescofo, based on timed traces comparisons and assume a given mixed score with a default tempo value.

score	::=	ϵ | evt score
evt	::=	e d s
s	::=	ϵ | act s
act	::=	d s al
al	::=	sync err
sync	::=	**@loose** | **@tight**
err	::=	**@local** | **@global**

Figure 4: Grammar of the handled AST-mixed score

2.2 Timed traces

A *timed trace* is a sequence of pairs $\langle a, t\rangle$ made of a symbol $a \in \mathcal{I} \cup \mathcal{O}$, and a *timestamp* $t \in \mathbb{R}^+$, either in physical or relative time.

A trace containing symbols exclusively in $\mathcal{I}$ (resp. $\mathcal{O}$) is called an *input trace* (resp. an *output trace*). We denote below $\mathcal{T}_{\mathsf{in}}$ (resp. $\mathcal{T}_{\mathsf{out}}$) the set of input (resp. output) traces with relative timestamps.

The *ideal trace* is the input trace consisting in the projection of all events in the mixed-score with their duration.

By definition of music performance, traces of real executions can be arbitrarily far from ideal traces: the tempo and delays can diverge from the written values (the musician adding her/his own expressiveness values), and moreover the musician can do errors during a performance (missing notes).

A *test case* is a pair $\langle t_{\mathsf{in}}, t_{\mathsf{out}}\rangle \in \mathcal{T}_{\mathsf{in}} \times \mathcal{T}_{\mathsf{out}}$ where $t_{\mathsf{in}} \in \mathcal{E}$ and $t_{\mathsf{out}} = \mathcal{S}(t_{\mathsf{in}})$. $t_{\mathsf{out}} = \mathcal{S}(t_{\mathsf{in}})$ is a simulation of t_{in} on the specification returning t_{out}. We note too $t'_{\mathsf{out}} = \mathrm{IUT}(t_{\mathsf{in}})$ an Antescofo execution with t_{in} as input trace and t'_{out} as monitored real trace.

3 Scenarios

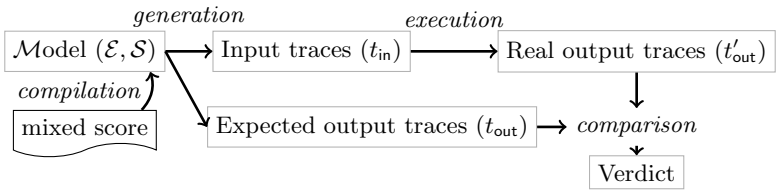


Figure 5: Score-based IMS testing workflow with traces

We recall that: The principle (Figure 5) is to execute a real *implementation under test* (IUT) in a testing framework. When the source code of the IUT is not known and only its input and output are observed, we call it *black-box* testing. In conformance *model-based testing* (MBT), a formal specification, or model $\mathcal{M}$ of the system is required to generate automatically some test data. This comprises input test data t_{in}, sent to the IUT, and theoretically expected output test data t_{out}, computed from t_{in} using $\mathcal{S}$. The latter t_{out} is then compared to the real output test data t'_{out}, obtained from the IUT when it receives t_{in}, in order to produce a test verdict.

3.1 Execution

The execution of a test case $\langle t_{\mathsf{in}}, t_{\mathsf{out}} \rangle$ consists in several tasks summarized in the following definition of conformance. First, we pass the events of t_{in} to the IMS Antescofo, respecting the timestamps. Second, we monitor the outcome of the IUT in an output trace $t'_{\mathsf{out}} = \mathrm{IUT}(t_{\mathsf{in}})$. Finally, we compare t'_{out} to $t_{\mathsf{out}} = \mathcal{S}(t_{\mathsf{in}})$. We define the *conformance* of the IUT to $\mathcal{S}$ *wrt* $\mathcal{E}$ as:

$\forall t_{\mathsf{in}} \in \mathcal{E},\ \mathrm{IUT}(t_{\mathsf{in}}) = \mathcal{S}(t_{\mathsf{in}})$.

This is a particular case of the relation $rtioco_{\mathbf{e}}$.

This definition makes sense only if the timestamps of t_{in}, t_{out} and t'_{out} are in the same time unit. We will show how this important issue is addressed in practice with several options for the conversion of all traces into physical time (thanks to the addition of tempo values).

One trace format is defining to link test softwares and Antescofo symbols. For input traces in use cases, we use the $\langle i, d, t \rangle$ input form (relations between $\langle a, t \rangle$ and $\langle i, d, T \rangle$ are explained section 4).

3.2 Tests use cases

Next sub sections present different use cases, implemented to test Antescofo. Each case has a proper handling of the MBT method which can be divided in three independent parts:

- The input, the execution and the conformance.

Input: From the inputs choice, the tester can test her/his mixed-score on a large scale or an exhaustive set of input traces, or just with a specific performance. The first case uses the automatic input trace generation with Uppaal and CoVer or the fuzz method (depending of the length of the mixed-score), and the latter case needs a single input trace or an audio file performance.

Execution: The execution scenario chosen, for a test, defines the black box bounds. Four scenarios are implemented testing from the RE only to the entire IUT in its MAX patch version.

Conformance: The conformance compares two traces. One (the compared trace) has to be an Antescofo output with physical timestamps and their corresponding tempos. So the second (expected trace) can be in relative or physical time. We can choose here whether the real and expected traces are conform or several real traces (obtained from different mixed scores) provoke the same Antescofo behavior (i.e all are conformed to one excepted trace).

To continue the three first use cases concern different scenarios of execution. The next presents the fuzz solution and the multiple conformances case is shown last.

3.2.1 Testing the RE

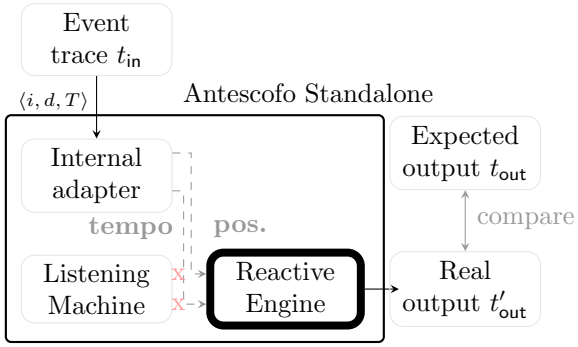


Figure 6: Testing scenario of Section 3.2.1.

This scenario is performed on a standalone version of Antescofo equipped with an internal *test adapter* module. The adapter iteratively reads one element $\langle i, d, T \rangle$ of a file containing t_{in}, converts d into a physical time value $d' = \frac{d.6.10^4}{T}$ and waits d' ms before sending i and T to the RE. More precisely, it does not physically wait, but instead notifies a *virtual clock* in the RE that the time has flown of d' ms. This way the test does not need to be executed in realtime but can be done in fast-fordward mode. This is very important for batch execution of huge lists of test cases.

The messages sent by RE are traced in t'_{out}, with timestamps in physical time (this functionality is built in the current RE). Finally, the timestamps in t_{out} are converted from relative time to physical time using the tempo values in t_{in}, in order to be comparable to t'_{out}. In this scenario, the IUT is the RE (the LM is idle).

3.2.2 Testing the RE with tempo detection

In this second scenario the tempo values T are not read in t_{in} by the adapter but instead inferred by the LM (the adapter is calling an appropriate method in LM). The rest of the scenario is similar to Section 3.2.1. The values of detected tempo are stored by the adapter and used later to convert the dates in the expected trace t_{out} from relative to physical time. In this case, the IUT is somehow the RE plus the part of the LM in charge of tempo inference.

3.2.3 Testing the whole system

This scenario is the most general. It is executed in a version of Antescofo embedded into MAX (as a MAX patch), using an adapter which is another

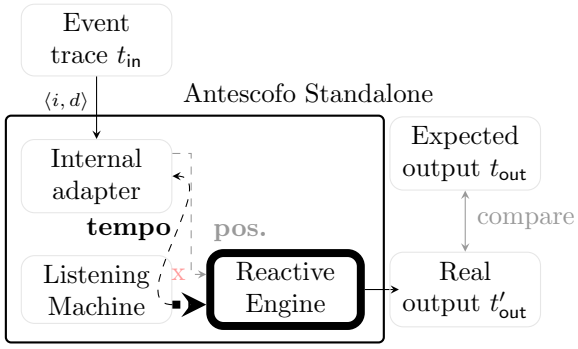


Figure 7: Testing scenario of Section 3.2.2.

MAX patch. The adapter iteratively reads triples $\langle i, d, T \rangle$ in a file containing t_{in}, and plays them as MIDI events, using the duration d converted to physical time with T. The audio stream generated is then sent to the LM, and the output of the RE is traced in t'_{out} as before.

Note that here, the RE uses the tempo values detected by the LM, which will differ from the tempo values in t_{in}. Therefore, the former are saved by the adapter (in MAX the detected tempo is available as an outlet of the antescofo~ patch), and used later to convert the dates in t_{out} from relative to physical time. In this realistic scenario, the IUT is the whole Antescofo system.

In an alternative scenario, the adapter ignores the tempo values in t_{in} and uses tempo detected by the LM, similarly to Section 3.2.2. Note that in these scenarios, the tests are executed in real-time and not in a fast-forward mode like in Sections 3.2.1.

3.2.4 Fuzz testing

The ***problem handled*** *here is the scalability.*
As explained before Uppaal tools as CoVer, used to generate a covering suite of input traces, can be overwhelming when too long mixed-scores are given in

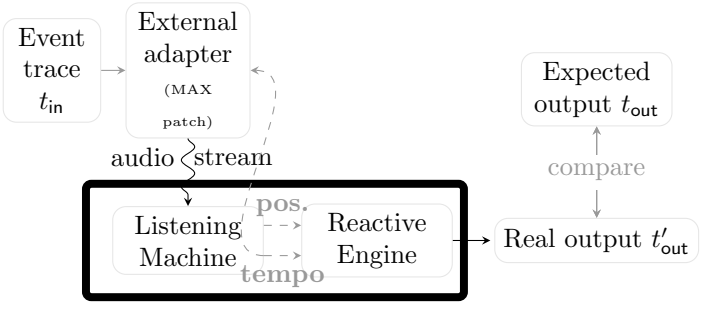


Figure 8: Testing scenarios of Section 3.2.3.

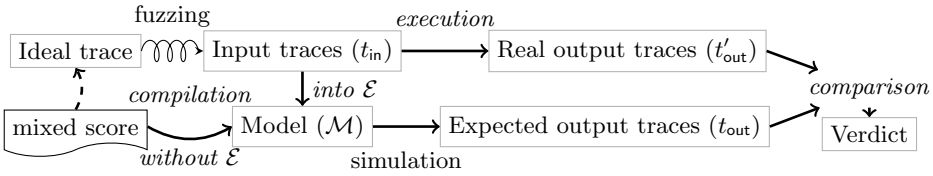


Figure 9: fuzz workflow

input.

Another generation can be done by *fuzzing the ideal trace* and set fuzzed performances in the environment ($\mathcal{E}$) of the model. The output after simulating $\mathcal{E}$ on the specification $\mathcal{S}$ of the mixed-score is the corresponding Antescofo reaction (the expected trace t_{out}).

Fuzzing can be performed by an external program which randomly chooses (in a given rate) a different value to the ideal trace. Note that inputs can be manually given by testers and/or composers (i.e via an Ascograph[1] extension). It will not change the rest of the test scenario since the performance is translated in Uppaal environment automaton $\mathcal{E}$.

3.2.5 Testing from an audio file

The ***problem handled*** *here is the model sub-language restriction.*
Our model doesn't cover the entire possibilities of the Antescofo DSL. However the verdict is done on the output traces of Antescofo.

This scenario needs a specifiable mixed-score named reference, an audio recorded file and a mixed-score folder. The idea is to derive once or more times a complex mixed-score (which can't be specified) and give a simpler version (reference) of the same score that can be specified (for a performance or a set). In a way it is an Antescofo specification.

[1] Ascograph is an Antescofo tool software to manage Antescofo and create a mixed-score graphically, `http://repmus.ircam.fr/mutant/ascograph`.

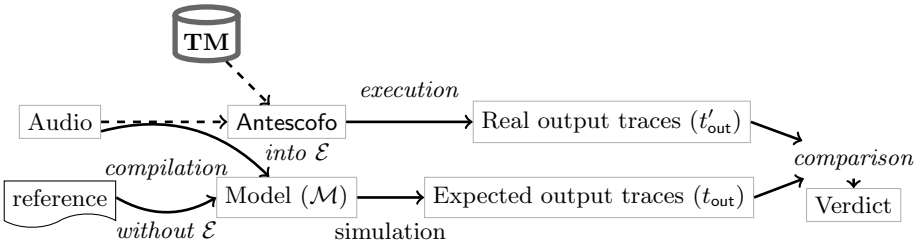


Figure 10: audio scenario case

The audio represents the input trace which will be converted into t_{in} and translated into $\mathcal{E}$. The reference is translated into the specification ($\mathcal{S}$) to generate the expected trace with $\mathcal{E}$. From here, the scenario takes the expected trace and for each mixed-score in the folder (named Tested Mixed-scores) checks the conformance of the result of Antescofo with the audio input.

Note that folder's mixed scores have no more DSL restriction and have to trace the same Antescofo behavior. The conversion of audio into t_{in} is done by passing the audio file to Antescofo which creates the corresponding input trace (that is detected by Antescofo).

4 Trace format and example

An example is depicted with the revisited Einspielung mixed-score. Only the event and action labels are added of the Nune's original version (we need single labels to identify throw severals programs each event and action of the tested mixed-score). An extract from Ascograph test-editor is shown Figure 11.

4.1 Input traces

An input trace is a list of input symbols, i.e mixed-score events. The format used during tests is a suite of lines representing the played events and containing:

- the identifier (notenum)
- the duration of the event (in beat)
- the current tempo during the duration (in BPM)
- the mixed-score label
- the list of notes playing for this event

The two latest fields are optional, and the tempo can be omitted for a relative input trace. As an example, Figure 12 depicts an absolute input trace, i.e an Einspielung performance.

In absolute cases, we represent the theoretic $\langle a, t\rangle$ with $\langle i, d, T\rangle$, i is the event identifier of Antescofo and d, T the duration and tempo used to compute the time stamp t in absolute time ($t = t_{-1} + d'$ with $d' = \frac{d.6.10^4}{T}$).

4.2 Output traces

Output traces are expected and real traces. To facilitate the understanding and further handles, we put too event detections in our traces (which aren't just output traces). Their general form is a list of time-stamped symbols.

4.2.1 Expected traces

An expected trace is a trace obtained by the $\mathcal{S}$pecification after a simulation of an input trace. Since it comes from the model we choose to keep all the time-stamps in relative time unit (beats). Events are kept for testing the detection or the difference between a simulation and an Antescofo trace that can be compared latter. The events are listed with:

- a keyword event

```
;               EINSPIELUNG I           E.Nunes

BPM 60.00

    antescofo-mess gamma -1.0


VARIANCE 0.25

; ---------- measure 1 beat 0.00 ---------

NOTE 6200 1/7   Mesure1



    Mac-1 ADC1-del 10   @name a1_0_0
    Mac-1 ADC2-del 0    @name a1_0_1
    Mac-1 ADC3-del 20   @name a1_0_2
    Mac-1 ADC4-del 0    @name a1_0_3
    Mac-1 ADC5-del 15   @name a1_0_4
    Mac-1 ADC6-del 30   @name a1_0_5


    Mac-1 ADC3-amp 12   @name a1_0_6
    Mac-1 ADC5-amp 12   @name a1_0_7

    Mac-1 Reson partie1 front amp -3 res 1. D @name a1_0_8 ;Mes-1-20
    Mac-1 rev-amp 0     @name a1_0_9
    Mac-1 Rev off       @name a1_0_10
    Mac-1 Filtres off   @name a1_0_11

    GFWD r1
    {
            Mac-1 ADC1 143 2 grun-l direct 10       @name a1_1_1
            Mac-1 ADC2 143 2 carre2 harmo 25        @name a1_1_2
            Mac-1 ADC3 143 2 trap-c harmo 25 0 -40 0.0.68.57 fac 1.5    @name a1_1_3
            Mac-1 ADC4 143 2 dim harmo -2467 amp -4 fac 1.5             @name a1_1_4
            Mac-1 ADC5 143 2 cresc harmo 1167 0 1200 0.0.68.57 fac 0.7  @name a1_1_5
            Mac-1 ADC6 143 2 grun-c harmo 3000                          @name a1_1_6

        1/7 Mac-1 ADC1 143 10 grun-l direct 10  @name a1_2_1
            Mac-1 ADC2 143 10 carre2 harmo -40  @name a1_2_2
            Mac-1 ADC3 143 10 trap-c harmo 20 0.0.68.57 fac 1.5     @name a1_2_3
            Mac-1 ADC4 143 10 dim harmo -2400 amp -4 fac 1.5        @name a1_2_4
            Mac-1 ADC5 143 10 cresc harmo 1267 0.0.68.57 fac 0.7    @name a1_2_5
            Mac-1 ADC6 143 10 grun-c harmo 3000                     @name a1_2_6

        1/7 Mac-1 ADC1 143 4 grun-l direct 10   @name a1_3_1
            Mac-1 ADC2 143 4 carre2 harmo 60    @name a1_3_2
            Mac-1 ADC3 143 4 trap-c harmo 130 0.0.68.57 fac 1.5 @name a1_3_3
            Mac-1 ADC4 143 4 dim harmo -2367 amp -4 fac 1.5     @name a1_3_4
            Mac-1 ADC5 143 4 cresc harmo 2333 0.0.68.57 fac 0.7 @name a1_3_5
            Mac-1 ADC6 143 4 grun-c harmo 3000                  @name a1_3_6

        1/7 Mac-1 ADC1 143 10 grun-l direct 10  @name a1_4_1
            Mac-1 ADC2 143 10 carre2 harmo 110  @name a1_4_2
            Mac-1 ADC3 143 10 trap-c harmo  70 0.0.68.57 fac 1.5    @name a1_4_3
100%    1:1     Operation complete
```

Figure 11: Einspielung test version extract in the Ascograph text-editor

- its label
- its duration (in beat)
- its time-stamp (in beat)
- its related tempo

And actions with:

- its time-stamp (in beat)
- its label

All the time-stamps are taken from the begging of the trace (a computation is needed to have the time elapsed since the nearest passed event to translate, with the corresponding tempo, the good absolute duration and time-stamp). An example is shown Figure 13.

4.2.2 Real traces

A real trace is a trace obtained with Antescofo test trace option and after an execution of an input trace for the tested mixed-score. Information is taken from Antescofo variables during the execution. An event has:

```
1 0.28656 64.2 " Mesure1 " 6200
2 0.14199 57.5 " e1_1 " ( 5800 6200 )
3 0.28999 60 " e1_2 " ( 6400 6200 )
4 0.443 60 " e1_3 " ( 7500 6200 )
5 0.546 60.8 " e1_4 " ( 6900 6200 )
6 0.697 58.7 " e1_5 " ( 6100 6200 )
7 0.896 61.5 " e1_6 " ( 6700 6200 )
9 1.207 76.1 " event2_1 " ( 6000 6800 )
10 1.341 86.5 " event2_2 " ( 6600 6800 )
11 1.44 58.9 " Mesure3 " 6200
12 1.671 55.2 " event3_1 " ( 7100 6200 )
13 1.757 56.1 " event3_2 " ( 6500 6200 )
14 1.855 56.9 " event3_3 " ( 6600 6200 )
15 1.995 86.5 " Mesure4 " ( 6800 6200 )
16 2.212 79 " e4_1 " ( 7300 6200 )
17 2.249 84.9 " e4_2 " ( 7700 6200 )
18 2.564 76.8 " e4_3 " ( 8300 6200 )
19 2.788 76.3 " e4_4 " ( 7600 6800 )
21 2.901 61.8 " Mesure5 " 6200
22 3.928 99.1 " Mesure6 " 68
23 4.314 105 " e6_1 " ( 5800 6800 )
25 4.466 105 " e6_3 " ( 7500 6800 )
```

Figure 12: An Einspielung input trace extract

- its identifier (notenum)
- its label
- its absolute time-stamp (gettime, the current time [but currently from a virtual clock])
- its dynamic score position (rnow)
- its current detected tempo (rt_tempo)

A message:

- its label
- its absolute time-stamp (gettime)
- its dynamic score position (rnow)
- the current tempo

Be **careful** the used time-stamps here (from gettime) are in ms. The real trace example is reported Figure 14

5 Verdict

For the moment the verdict was only used for the development of the test method but aims at being a complete representation of test information.

The verdict version reports in a file:

- the version and options of the test
- the input trace linked
- a table of real/expected symbols
- a verdict, ok or ko if the test pass or not

In the table is annotated for each symbol the label, its absolute time-stamp and its relative one (in []). The current tempo for events *, missed symbols (x for messages, ^{o} for events). And the delta when the difference of the real/expected time-stamps is greater than the test tolerance.

Figure 15 and 16 present two extracts of the verdict of Einspielung (the beggining and the end).

```
START of traces

EVENT e0 0.28656 (at 0) tempo=64.2
0 a0
0 a1
0 a2
0 a3
0 a4
0 a5
0 a6
0 a7
0 a8
0 a9
0 a10
0 a11
0 a12
0 a13
0 a14
0 a15
0 a16
0 a17
0.14285 a18
0.14285 a19
0.14285 a20
0.14285 a21
0.14285 a22
0.14285 a23
0.2857 a24
0.2857 a25
0.2857 a26
0.2857 a27
0.2857 a28
0.2857 a29
EVENT e1 0.14199 (at 0.28656) tempo=57.5
0.42855 a30
0.42855 a31
0.42855 a32
0.42855 a33
0.42855 a34
0.42855 a35
EVENT e2 0.28999 (at 0.42855) tempo=60
0.5714 a36
0.5714 a37
```

Figure 13: An Einspielung expected trace extract

```
 gamma -1 0 0 60
a1_0_0 0 0 64.1999964029791
a1_0_1 0 0 64.1999964029791
a1_0_2 0 0 64.1999964029791
a1_0_3 0 0 64.1999964029791
a1_0_4 0 0 64.1999964029791
a1_0_5 0 0 64.1999964029791
a1_0_6 0 0 64.1999964029791
a1_0_7 0 0 64.1999964029791
a1_0_8 0 0 64.1999964029791
a1_0_9 0 0 64.1999964029791
a1_0_10 0 0 64.1999964029791
a1_0_11 0 0 64.1999964029791
r1 0 0 64.1999964029791
a1_1_1 0 0 64.1999964029791
a1_1_2 0 0 64.1999964029791
a1_1_3 0 0 64.1999964029791
a1_1_4 0 0 64.1999964029791
a1_1_5 0 0 64.1999964029791
a1_1_6 0 0 64.1999964029791
1 Mesure1 0 0 64.1999964029791

a1_2_1 0.133511355945042 0.142857142857143 64.1999964029791
a1_2_2 0.133511355945042 0.142857142857143 64.1999964029791
a1_2_3 0.133511355945042 0.142857142857143 64.1999964029791
a1_2_4 0.133511355945042 0.142857142857143 64.1999964029791
a1_2_5 0.133511355945042 0.142857142857143 64.1999964029791
a1_2_6 0.133511355945042 0.142857142857143 64.1999964029791
a1_3_1 0.267022711890084 0.142857142857143 64.1999964029791
a1_3_2 0.267022711890084 0.142857142857143 64.1999964029791
a1_3_3 0.267022711890084 0.142857142857143 64.1999964029791
a1_3_4 0.267022711890084 0.142857142857143 64.1999964029791
a1_3_5 0.267022711890084 0.142857142857143 64.1999964029791
a1_3_6 0.267022711890084 0.142857142857143 64.1999964029791
2 e1_1 0.26781309684269 0.142857149243355 57.5000005712112

a1_4_1 0.41597657510356 0.284847150653899 60
a1_4_2 0.41597657510356 0.284847150653899 60
```

Figure 14: An Einspielung real trace extract

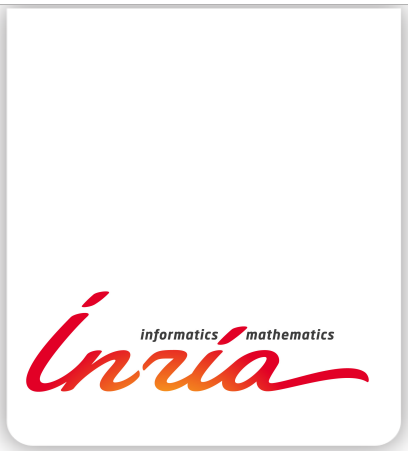
```
Tr2Score V4 (New test procedure)
option verification Absolute
... Verification on air ... Nunes_Einspielung1/Mesure14/Einspielung_14-obtained/Einspielung_
start of tests
Mesure1 --> 0.28656 --> e1_1 --> 0.14199 --> e1_2 --> 0.28999 --> e1_3 --> 0.443 --> e1_4 --
```

	Antescofo Trace		Expected Trace		
	TimeStamp	[Estimate Beat]	TimeStamp	[Beat]	
	a1_0_0	0 [0]	a0	0 [0]	
	a1_0_1	0 [0]	a1	0 [0]	
	a1_0_2	0 [0]	a2	0 [0]	
	a1_0_3	0 [0]	a3	0 [0]	
	a1_0_4	0 [0]	a4	0 [0]	
	a1_0_5	0 [0]	a5	0 [0]	
	a1_0_6	0 [0]	a6	0 [0]	
	a1_0_7	0 [0]	a7	0 [0]	
	a1_0_8	0 [0]	a8	0 [0]	
	a1_0_9	0 [0]	a9	0 [0]	
	a1_0_10	0 [0]	a10	0 [0]	
	a1_0_11	0 [0]	a11	0 [0]	
	a1_1_1	0 [0]	a12	0 [0]	
	a1_1_2	0 [0]	a13	0 [0]	
	a1_1_3	0 [0]	a14	0 [0]	
	a1_1_4	0 [0]	a15	0 [0]	
	a1_1_5	0 [0]	a16	0 [0]	
	a1_1_6	0 [0]	a17	0 [0]	
*	Mesure1	0 [0]	e0	0 [0]	* T: 64.2 BPM
	a1_2_1	0.134 [0.143]	a18	0.134 [0.143]	
	a1_2_2	0.134 [0.143]	a19	0.134 [0.143]	
	a1_2_3	0.134 [0.143]	a20	0.134 [0.143]	
	a1_2_4	0.134 [0.143]	a21	0.134 [0.143]	
	a1_2_5	0.134 [0.143]	a22	0.134 [0.143]	
	a1_2_6	0.134 [0.143]	a23	0.134 [0.143]	
	a1_3_1	0.267 [0.286]	a24	0.267 [0.286]	
	a1_3_2	0.267 [0.286]	a25	0.267 [0.286]	
	a1_3_3	0.267 [0.286]	a26	0.267 [0.286]	
	a1_3_4	0.267 [0.286]	a27	0.267 [0.286]	
	a1_3_5	0.267 [0.286]	a28	0.267 [0.286]	
	a1_3_6	0.267 [0.286]	a29	0.267 [0.286]	
*	e1_1	0.26781 [0.257]	e1	0.26781 [0.287]	* T: 57.5 BPM
	a1_4_1	0.416 [0.399]	a30	0.416 [0.429]	
	a1_4_2	0.416 [0.399]	a31	0.416 [0.429]	

Figure 15: An Einspielung verdict extract (first lines)

```
*  Mesure13  303.61 [ 374]          e62  303.61 [ 372]          * T: 60.1 BPM
    a13_2_1     304 [ 374]         a357     304 [ 372]          x delta: 0.000856
    a13_2_2     304 [ 374]         a358     304 [ 372]          x delta: 0.000856
    a13_2_3     304 [ 374]         a359     304 [ 372]          x delta: 0.000856
    a13_3_1     304 [ 374]         a360     304 [ 372]          x delta: 0.000713
    a13_3_2     304 [ 374]         a361     304 [ 372]          x delta: 0.000713
    a13_3_3     304 [ 374]         a362     304 [ 372]          x delta: 0.000713
    a13_4_1     304 [ 374]         a363     304 [ 373]          x delta: 0.00057
    a13_4_2     304 [ 374]         a364     304 [ 373]          x delta: 0.00057
    a13_4_3     304 [ 374]         a365     304 [ 373]          x delta: 0.00057
    a13_5_1     304 [ 374]         a366     304 [ 373]          x delta: 0.000428
    a13_5_2     304 [ 374]         a367     304 [ 373]          x delta: 0.000428
    a13_5_3     304 [ 374]         a368     304 [ 373]          x delta: 0.000428
    a13_6_1     304 [ 375]         a369     304 [ 373]          x delta: 0.000285
    a13_6_2     304 [ 375]         a370     304 [ 373]          x delta: 0.000285
    a13_6_3     304 [ 375]         a371     304 [ 373]          x delta: 0.000285
    a13_7_1     304 [ 375]         a372     304 [ 373]          x delta: 0.000143
    a13_7_2     304 [ 375]         a373     304 [ 373]          x delta: 0.000143
*     e13_1  318.51 [ 388]          e63  318.51 [ 387]          * T: 58.4 BPM
*     e13_2  334.51 [ 403]          e64  334.51 [ 403]          * T: 56.2 BPM
*     e13_3  351.53 [ 420]          e65  351.53 [ 419]          * T: 57.4 BPM
*     e13_4  367.53 [ 435]          e66  367.53 [ 434]          * T: 58.5 BPM
*     e13_5     383 [ 451]          e67     383 [ 449]          * T: 60.7 BPM
*     e13_6  397.85 [ 465]          e68  397.85 [ 464]          * T: 58.1 BPM
    a14_0_1     415 [ 482]         a374     415 [ 480]
    a14_0_2     415 [ 482]         a375     415 [ 480]
    a14_0_3     415 [ 482]         a376     415 [ 480]
    a14_0_4     415 [ 482]         a377     415 [ 480]
    a14_1_1     415 [ 482]         a378     415 [ 480]
    a14_2_1     415 [ 482]         a385     415 [ 480]
*  Mesure14  414.66 [ 495]          e69  414.66 [ 480]          * T: 105 BPM
    a14_1_2     415 [ 496]         a379     415 [ 481]
    a14_2_2     415 [ 496]         a386     415 [ 482]
    a14_1_3     416 [ 497]         a380     416 [ 482]
    a14_2_3     416 [ 497]         a387     416 [ 483]
    a14_1_4     416 [ 498]         a381     416 [ 483]
    a14_1_5     417 [ 499]         a382     417 [ 484]
    a14_2_4     417 [ 499]         a388     417 [ 484]
    a14_1_6     418 [ 500]         a383     418 [ 485]
    a14_2_5     418 [ 500]         a389     418 [ 486]
    a14_1_7     418 [ 501]         a384     418 [ 486]
*       e15  424.13 [ 509]          e70  424.13 [ 497]          * T: 90.9 BPM
*       END  439.86 [ 533]          e71  439.86 [ 521]          * T: 92.5 BPM
|-----------------------------------------------------------------|
Error :: Test KO
```

Figure 16: An Einspielung verdict extract (last lines)



RESEARCH CENTRE
PARIS – ROCQUENCOURT

Domaine de Voluceau, - Rocquencourt
B.P. 105 - 78153 Le Chesnay Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399